

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**ADVANCING ROC CURVE MODELING WITH
INFORMATION COMPLEXITY FOR
EVALUATING THE PERFORMANCE OF
MACHINE LEARNING ALGORITHMS**

by
Aylin GÖÇÖĞLU

February, 2025

İZMİR

ADVANCING ROC CURVE MODELING WITH INFORMATION COMPLEXITY FOR EVALUATING THE PERFORMANCE OF MACHINE LEARNING ALGORITHMS

**A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of
Philosophy in Statistics, Statistics Program**

**by
Aylin GÖÇÖĞLU**

February, 2025

İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**ADVANCING ROC CURVE MODELING WITH INFORMATION COMPLEXITY FOR EVALUATING THE PERFORMANCE OF MACHINE LEARNING ALGORITHMS**” completed by **AYLİN GÖÇÖĞLU** under supervision of **PROF. DR. NESLİHAN DEMİREL** and **PROF. DR. HAMPARSUM BOZDOĞAN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....
Prof. Dr. Neslihan DEMİREL

Supervisor

.....
Assoc. Prof. Engin YILDIZTEPE

Thesis Committee Member

.....
Prof. Dr. Mustafa Alper SELVER

Thesis Committee Member

.....
Prof. Dr. Gözde ULUTAGAY

Examining Committee Member

.....
Prof. Dr. Hayal BOYACIOĞLU

Examining Committee Member

.....
Prof. Dr. Abdullah SEÇGİN
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my deepest and most heartfelt gratitude to my advisor, Prof. Dr. Neslihan DEMİREL, who has been a cornerstone of my academic journey. I am truly grateful for her unwavering belief in me, her endless support, and her exceptional guidance. Her guidance has not only shaped my academic journey but has also illuminated my path in personal life, providing me with a more positive perspective. I feel honored and privileged to be a student of her.

I would like to extend my gratitude and thanks to Prof. Dr. Hamparsum BOZDOĞAN for hosting me for one year as a Visiting Doctoral Scholar at the Business Analytics and Statistics Department at the University of Tennessee, in Knoxville, Tennessee. As a leading scholar in the field of statistics, his invaluable guidance has greatly expanded my research and understanding. The opportunity to collaborate on ICOMP-ROC with him has been one of the most rewarding experiences of my career, and I am honored to have been part of this work. Beyond academia, I am deeply grateful for the warmth and kindness he showed me during my year in the United States, ensuring that I never felt away from home.

I want to thank Assoc. Prof. Dr. Engin YILDIZTEPE and Prof. Dr. Alper SELVER, who are on the thesis meeting committee, shared their knowledge and suggestions about my work and guided me with their contributions. I also take this opportunity to express gratitude to all of the Department of Statistics members who have contributed to my academic journey. Moreover, I would like to express my sincere gratitude to Prof. Dr. Nur MOLLAOĞLU and Asst. Prof. Özgün YILDIRIM from the Division of Oral and Maxillofacial Surgery, Faculty of Dentistry at Gazi University for sharing with me the dataset used in the application chapter of my thesis.

I am deeply honored to have been supported by the Higher Education Council of Turkey (YÖK) 100/2000 Ph.D. Scholarship Program throughout the four years. I would also like to extend my heartfelt gratitude to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for granting me the opportunity to continue my doctoral research for one year at the University of Tennessee, in Knoxville, Tennessee.

I am deeply grateful to my dear friend Ayça ÖLMEZ for her unwavering support in every aspect of my life, her incredible patience, and for making everything more enjoyable. Her unique creativity always succeeds in revealing beauties that I would never be able to notice without her. I truly appreciate our bond, which goes beyond friendship. I sincerely thank my dear friend Yusuf Can SEVİL for always being by my side with his strong work discipline and for the many brainstorming sessions where we found solutions together. I sincerely thank my dear friend Onurcan KÖKEN for always being by my side, making even the furthest miles feel close with his wisdom and friendship. I would like to express my deepest gratitude to my dear Cengiz Mehmet ALBOYACI for always broadening my perspective through our countless deep conversations. His unwavering belief in me, endless patience, and constant encouragement have given me the strength to succeed.

I am deeply grateful to my beloved family for their unwavering love, endless support, and boundless patience. Their love has always been my greatest refuge and my strongest source of strength. This achievement is as much theirs as it is mine.

I extend my deepest respect to Mustafa Kemal ATATÜRK, whose dedication to education and science laid the foundation for independence and modernity, making this journey possible and inspiring us all.

Aylin GÖÇOĞLU

ADVANCING ROC CURVE MODELING WITH INFORMATION COMPLEXITY FOR EVALUATING THE PERFORMANCE OF MACHINE LEARNING ALGORITHMS

ABSTRACT

Performance metrics are measures of performance that can be used to evaluate how well a model makes accurate predictions or classifications. However, there is no single measure, since each performance metric emphasizes a different classification aspect. Model selection procedures based on information criteria offer a quantitative measure that balances model complexity with goodness of fit, providing a better alternative to classical approaches. In this thesis, we introduce and develop a novel Information Complexity-Receiver Operating Characteristic, abbreviated as ICOMP-ROC, criterion approach to fit and study the performance of ROC curve models. We construct and derive the Universal ROC (UROC) for the combination of sixteen Bi-distributional ROC models to choose the best Bi-distributional ROC by minimizing the ICOMP-ROC criterion. We conduct a large-scale Monte Carlo simulation study using the sixteen Bi-distributional ROC models with the Weibull-Gamma and Normal-Normal pairs as the pseudo-true ROC models. We report the frequency of hits of the ICOMP-ROC criterion, showing its remarkable recovery rate. In addition to Bi-distributional fitting, we consider orthopantomography (OPG) of individuals who underwent dental implant treatment dataset to study the performance of the common performance metrics and the ICOMP-ROC criterion using several machine learning (ML) classification algorithms. The dataset was divided into balanced and imbalanced subsets to evaluate model performance. Each subset was analyzed using both the full dataset and splitting it into training (80%), testing (20%) sets. This dual approach allowed for a comprehensive assessment of model robustness in both controlled and realistic conditions. We use the Genetic Algorithm (GA) to reduce the dimension of the dataset to choose the best subset of the features to study and compare the performance of the newly proposed ICOMP-ROC criterion along with the traditional performance

metrics. The choice of a suitable metric is contingent upon not just the ML model used, but it also depends upon the complexity and high dimensionality of the input datasets, since the traditional performance metrics give different results and have inherent limitations. Our numerical results show the consistency and reliability of the ICOMP-ROC criterion over the traditional ROC performance metrics as a clever model selection criterion to choose the best fitting Bi-distributional ROC model and the best classification algorithm among the ones considered. This shows the utility and the versatility of our newly proposed approach in ROC curve modeling that integrates and robustifies currently used procedures.

Keywords: Information complexity, ROC curve, model selection, ICOMP-ROC, Bi-distributional ROC, Universal ROC, genetic algorithm, performance of classifiers.

MAKİNE ÖĞRENİMİ ALGORİTMALARININ PERFORMANSINI DEĞERLENDİRMEK İÇİN BİLGİ KARMAŞIKLIĞIYLA ROC EĞRİSİ MODELİNİN GELİŞTİRİLMESİ

ÖZ

Performans metrikleri, bir modelin tahmin veya sınıflandırma doğruluğunu değerlendirmek için kullanılan ölçütlerdir. Ancak, her performans ölçütü farklı bir sınıflandırma yönünü vurguladığı için tek bir ölçüt yoktur. Bilgi kriterlerine dayalı model seçim prosedürleri, model karmaşıklığını uyum iyiliğiyle dengeleyen niceliksel bir ölçüm sunarak klasik yaklaşımlara kıyasla daha iyi bir alternatif sağlar. Bu tezde, Alıcı işlem karakteristiği (ROC) eğrisi modellerinin performansını değerlendirmek ve incelemek amacıyla ICOMP-ROC olarak kısaltılan yeni bir Bilgi Karmaşıklığı-Alıcı işlem karakteristiği yaklaşımı tanıtılmış ve geliştirilmiştir. ICOMP-ROC kriterini minimize ederek en iyi iki değişkenli dağılım ROC modelini seçmek amacıyla, on altı iki değişkenli dağılım ROC modelinin kombinasyonuna dayalı Evrensel ROC (UROC) türetilmiştir. Weibull-Gamma ve Normal-Normal dağılım çiftlerini sözde-gerçek dağılımlar olarak kabul ederek, on altı iki değişkenli dağılım ROC modelinden doğru olanları belirlemek amacıyla büyük ölçekli bir Monte Carlo simülasyon çalışması yürütülmüştür. ICOMP-ROC kriterinin başarı frekansını rapor edilerek, simülasyon boyunca doğru modeli kaç kez seçtiği hesaplanmış ve önemli bir doğrulama oranı elde edilmiştir. İki değişkenli dağılıma ek olarak, diş implantı tedavisi gören bireylerin ortopantomografi (OPG) veri seti kullanılarak yaygın kullanılan performans metrikleri ve ICOMP-ROC kriterinin performansı çeşitli makine öğrenimi (ML) sınıflandırma algoritmaları kullanılarak değerlendirilmiştir. Veri kümesi, model performansını değerlendirmek amacıyla dengeli ve dengesiz alt kümelere ayrılmıştır. Her alt küme, hem tam veri kümesi kullanılarak hem de eğitim (%80), test (%20) olarak bölünerek analiz edilmiştir. Bu çift yönlü yaklaşım, modelin hem kontrollü hem de gerçekçi koşullardaki dayanıklılığını kapsamlı bir şekilde değerlendirmeye olanak sağlamıştır. Veri setinin boyutunu azaltmak için Genetik Algoritma (GA) kullanarak en iyi özellik alt kümesi seçilmiş ve yeni önerilen

ICOMP-ROC kriteri ile geleneksel performans metriklerinin performansları karşılaştırılmıştır. Uygun bir performans metriğinin seçimi yalnızca kullanılan makine öğrenimi (ML) modeline bağlı değildir; aynı zamanda giriş veri setlerinin karmaşıklığı ve yüksek boyutluluğu da önemli bir rol oynar. Geleneksel performans metrikleri farklı sonuçlar verebilir ve yapısal sınırlamalara sahiptir. Sayısal sonuçlarımız, ICOMP-ROC kriterinin geleneksel ROC performans metriklerine kıyasla daha tutarlı ve güvenilir olduğunu göstermektedir. Bu kriter, en iyi uyum sağlayan iki değişkenli dağılım ROC modelini ve değerlendirilen algoritmalar arasındaki en başarılı sınıflandırma modelini seçmede etkili bir yöntem sunmaktadır. Bu bulgular, mevcut yöntemleri bütünleştirerek daha sağlam hale getiren yeni önerilen yaklaşımımızın ROC eğrisi modellemesinde kullanışlı ve esnek bir çözüm sunduğunu ortaya koymaktadır.

Anahtar kelimeler: Bilgi karmaşıklığı, ROC eğrisi, model seçimi, ICOMP-ROC, İki değişkenli dağılım ROC, Evrensel ROC, genetik algoritma, sınıflandırıcıların performansı.

CONTENTS

	Page
Ph.D. THESIS EXAMINATION RESULT FORM	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	v
ÖZ.....	vii
LIST OF FIGURES	xii
LIST OF TABLES.....	xiii
ABBREVIATIONS	xiv
 CHAPTER ONE – INTRODUCTION.....	 1
 CHAPTER TWO – INFORMATION COMPLEXITY AND ITS BACKGROUND.....	 4
2.1 Kullback-Liebler Distance and Fisher Information.....	5
2.2 Akaike Information Criterion (AIC).....	6
2.3 Information Complexity Criterion (ICOMP)	7
2.3.1 ICOMP (IFIM)	9
 CHAPTER THREE – A NEWLY PROPOSED ICOMP-ROC CRITERION... 10	
3.1 Universal ROC Curve.....	10
3.2 A Newly Proposed ICOMP-ROC Criterion	11
3.3 Large Scale Monte Carlo Simulation Studies.....	14
3.3.1 Results of Monte Carlo Simulation Study	18
3.3.1.1 Case 1: Weibull-Gamma Bi-Distribution Pair	18
3.3.1.2 Case 2: Normal-Normal Bi-Distribution Pair	21
 CHAPTER FOUR – FEATURE SELECTION WITH THE GENETIC ALGORITHM.....	 26

4.1 Overview of Genetic Algorithm	26
4.2 Basic Terminology	27
4.3 GA Operators.....	28
4.3.1 Selection	28
4.3.2 Crossover	29
4.3.3 Mutation.....	31
4.3.4 Termination Criteria	31
 CHAPTER FIVE – OVERVIEW OF MACHINE LEARNING CLASSIFIERS	34
5.1 Logistic Regression Classifier	35
5.2 Support Vector Machine Classifier.....	36
5.3 Naive Bayes Classifier.....	37
5.4 K-Nearest Neighbors Classifier	39
5.5 Decision Tree Classifier	40
 CHAPTER SIX – APPLICATION OF ICOMP-ROC TO DENTAL IMPLANT DATA CLASSIFICATION	42
6.1 Data Acquisition	44
6.2 Pre-Processing	45
6.3 Feature Extraction	46
6.4 Datasets and Feature Selection with GA.....	46
6.5 Machine Learning Classification Algorithms.....	48
6.6 Performance Metrics and Information-Based Criteria for Classification	49
6.7 Main Computational Results	51
6.7.1 Results of case 1: Balanced Dataset with Full Data	51
6.7.2 Results of case 2: Balanced Dataset with Train/Test Split Data	53
6.7.3 Results of case 3: Imbalanced Dataset with Full Data.....	55
6.7.4 Results of case 4: Imbalanced Dataset with Train/Test Split Data	58
 CHAPTER SEVEN – CONCLUSIONS AND DISCUSSION.....	61

REFERENCES.....	64
-----------------	----



LIST OF FIGURES

	Page
Figure 3.1 Demonstration of real (a) and fitted (b) negative and positive class for Weibull-Gamma Bi-distribution pair	18
Figure 3.2 The ROC curves for the sixteen Bi-distribution pairs for pseudo-true Weibull-Gamma Bi-distribution pair	20
Figure 3.3 Demonstration of real (a) and fitted (b) negative and positive class for the Normal-Normal Bi-distribution pair	22
Figure 3.4 The ROC curves for the sixteen Bi-distribution pairs for pseudo-true Normal-Normal Bi-distribution pair	24
Figure 4.1 Single point crossover	30
Figure 4.2 Two point crossover	30
Figure 4.3 Uniform crossover	30
Figure 6.1 (a) Dental Implant Structure (LTD. (2024)) and (b) process of dental implant insertion (Burbank (2024))	42
Figure 6.2 Flowchart of the process	44
Figure 6.3 Anonymised OPG examples	45
Figure 6.4 Illustration of Confusion Matrix (a) and ROC Curve (b)	49
Figure 6.5 Comparison of classification methods ROC curves for balanced dataset with full data	53
Figure 6.6 Comparison of classification methods ROC curves for dataset with train/test split data	55
Figure 6.7 Comparison of classification methods ROC curves for imbalanced dataset with full data	57
Figure 6.8 Comparison of classification methods ROC curves for imbalanced dataset with train/test split data	60

LIST OF TABLES

	Page
Table 3.1 Probability distributions	15
Table 3.2 Cumulative distribution functions	16
Table 3.3 Inverse cumulative distribution functions	16
Table 3.4 Results obtained by running the Monte Carlo simulation once for the Weibull- Gamma Bi-distribution pair	19
Table 3.5 Frequency of success for the Monte Carlo simulation study for the Weibull- Gamma Bi-distribution pair	21
Table 3.6 Results obtained by running the Monte Carlo simulation once for the Normal- Normal Bi-distribution pair	23
Table 3.7 Frequency of success for the Monte Carlo simulation study for Normal-Normal Bi-distribution pair	25
Table 4.1 GA operational parameters	32
Table 6.1 Number of OPGs and implants of implant types.....	46
Table 6.2 GA parameters	48
Table 6.3 Performance Metrics	50
Table 6.4 Comparison of performance metrics and model selection criteria for balanced dataset with full data	52
Table 6.5 Comparison of performance metrics and model selection criteria for balanced dataset with train/test split data	54
Table 6.6 Comparison of performance metrics and model selection criteria for imbalanced dataset with full data	56
Table 6.7 Comparison of performance metrics and model selection criteria for imbalanced dataset with train/test split data	58

ABBREVIATIONS

ROC	: Receiver Operating Characteristic
TPR	: True Positive Rate
FPR	: False Positive Rate
AUC	: Area Under Curve
UROC	: Universal Receiver Operating Characteristic
ML	: Machine Learning
ICOMP-ROC	: Information Complexity - Receiver Operating Characteristic
GA	: Genetic Algorithm
ICOMP	: Information Complexity
AIC-ROC	: Akaike Information Criterion - Receiver Operating Characteristic
LR	: Logistic Regression
SVM	: Support Vector Machine
NB	: Naive Bayes
KNN	: K-Nearest Neighbors
DT	: Decision Tree
OPG	: Orthopantomography
AIC	: Akaike Information Criterion
TIC	: Takeuchi Information Criterion
GAIC	: Generalized Akaike Information Criterion
KL	: Kullback-Leiber
BIC	: Bayesian Information Criterion
HQC	: Hannan-Quinn Information Criterion

CAIC	: Consistent Akaike Information Criterion
CAICF	: Consistent Akaike Information Criterion with Fisher Information
FIM	: Fisher Information Matrix
MLE	: Maximum Likelihood Estimate
IFIM	: Inverse Fisher Information Matrix
ICOMP (IFIM)	: Information Complexity (Inverse Fisher Information Matrix)
CDF	: Cumulative Distribution Function
PDF	: Probability Density Function
EV	: Extreme Value
GEV	: Generalized Extreme Value
GP	: Generalized Pareto
ICDF	: Inverse Cumulative Distribution Function
GKQ	: Gaus-Kronrod Quadrature
RBF	: Radial Basis Function
SMO	: Sequential Minimal Optimization
MAP	: Maximum A Posteriori
CART	: Classification and Regression Trees
DWT	: Discrete Wavelet Transform
TP	: True Positive
TN	: True Negative
FP	: False Positive
FN	: False Negative
KDE	: Kernel Density Estimation



CHAPTER ONE

INTRODUCTION

A Receiver Operating Characteristic (ROC) curve is a unit square plot for simultaneously displaying the trade off between the True Positive Rate (TPR), which is the probability that the model correctly predicts the positive class, and the False Positive Rate (FPR), which is the probability that the model incorrectly predicts the positive class, for a binary classifier at different classification thresholds. Therefore, ROC curve is one of the widely used classification tools that helps in assessing the performance of the diagnostic tests and comparing these tests using accuracy, sensitivity, specificity, and the Area Under Curve (AUC).

In the vast literature of ROC, in terms of Bi-distributions, it is observed that the most commonly used Bi-distributional ROC is the Bi-Normal ROC model. Bi-Normal ROC model assumes that the random variables X_1 and X_2 are distributed according to normal (Gaussian) distributions with their respective means and variances Egan (1975). Numerous studies have applied the Bi-Normal model, including the recent papers in medical research by Shibata et al. (2009) and Wei et al. (2009). However, in most scientific fields of investigation, there is no guarantee that the data will follow a normal distribution. Hence, there can be substantial bias in the pointwise estimates of the estimated ROC curve, which in turn can create inaccurate thresholds for the final decision rule. This situation has created a new opportunity and a need for new research direction in ROC models when the normal distributional assumption is not achievable. In terms of other Bi-distributions, some notable studies include Bi-Beta (Zou et al. (1997)), Bi-Exponential (Tang et al. (2010)), Bi-Gamma (Hussain (2012)), and Bi-Weibull (Vardhan et al. (2012)) to mention a few. Despite many years of scientific effort, achieving a comprehensive generalization has remained elusive and continues to be an open area of research pursued by many. For example, for the Bi-distributional ROC curve model, Gneiting & Walz (2022) and Gneiting & Vogel (2022) proposed the Universal ROC (UROC) curve, which represents a generalized form of the ROC curve model that addresses several

shortcomings in the literature. The UROC curve combines multiple individual ROC curves into a single curve, weighted based on class configurations. These configurations are determined by the unique values of the outcome. This process is performed in a well-defined manner, ensuring accurate representation of the overall model performance.

For high-dimensional datasets, the success of statistical modeling techniques depends on identifying and selecting the most informative predictor variables. High-dimensional data often have many unimportant variables (or features) and a small number of relevant variables. The presence of irrelevant variables reduces the performance of the classification machine learning (ML) algorithms. Therefore, it is crucial to identify and choose the relevant variables when datasets contain numerous explanatory variables and there is limited prior knowledge about their importance. Because of this, choosing the most relevant variables is a challenge for researchers. When the goal is classification of the high-dimensional datasets and to minimize the number of false positives and false negatives while maximizing the number of true positives and true negatives, model selection procedures based on information criteria offer a superior alternative to classical approaches.

Therefore, in summary, our objective and contributions in this thesis are several fold.

- To address and resolve the existing problems in currently practiced ROC curve modeling, for the first time, we introduce and develop a new Information Complexity-Receiver Operating Characteristic (ICOMP-ROC) criterion.
- Using the UROC curve formalism, we generalize the Bi-distributional ROC curve model to inventory of Bi-distributions and show how to choose the best fitting ROC curve model with the ICOMP-ROC criterion.
- The performance metrics fall short because they do not simultaneously consider model complexity, especially in complex, high-dimensional datasets. To robustify the performance metrics of classifiers in high dimensions, we use the genetic algorithm (GA) in selecting the best subset of features with

ICOMP-ROC, which provides a comprehensive framework for evaluating model performance and complexity at the same time, thereby facilitating more informed, reliable, and interpretable results in the decision-making process.

The structure of the thesis is as follows: Chapter 2 discusses the general background of the Information Complexity (ICOMP) criterion. Chapter 3 includes the definition of the Universal ROC Curve (UROC) and introduces the new ICOMP-ROC criterion along with its derived form for the bivariate normal (Gaussian) distribution in matrix form. An inventory of Bi-distributional ROC curve models using different combinations of probability distributions is presented, with the Weibull-Gamma and Normal-Normal (Bi-Normal) serving as pseudo-true pairs of Bi-distributional models. Results from extensive Monte Carlo simulations under two pseudo-true distributions and a wide range of symmetric and asymmetric Bi-distributions commonly fitted to real datasets are provided, along with AUC, Akaike Information Criterion - Receiver Operating Characteristic (AIC-ROC), and ICOMP-ROC values for all positive and negative classes of distributions. The frequency of success in Monte Carlo simulation studies is also presented. Chapter 4 describes the use of the Genetic Algorithm (GA) for selecting an optimal subset of features to reduce dimensionality and improve the classification accuracies of Machine Learning (ML) classification algorithms. Chapter 5 examines several ML classification algorithms, including Logistic Regression (LR), Support Vector Machines (SVM), Naive Bayes (NB), K-Nearest Neighbors (KNN), and Decision Trees (DT). Chapter 6 focuses on a real numerical example and presents the results of an analysis conducted on orthopantomography (OPG) data from individuals undergoing dental implant treatment. Preprocessing, feature extraction, traditional performance metrics, and the computation of the newly proposed ICOMP-ROC model selection criteria for classification problems are covered. Computational results on real balanced and imbalanced OPG datasets are presented, and the performance of the proposed ICOMP-ROC criteria is compared with other traditional performance metrics. Finally, Chapter 7 provides a summary of findings and conclusions.

CHAPTER TWO

INFORMATION COMPLEXITY AND ITS BACKGROUND

Model selection is the process of finding the best model from a subset of existing models by revealing which of the explanatory variables influences the response variable. Determining the best model is one of the biggest problems in cases where there are many explanatory variables and there is no or insufficient prior information about the model. For this reason, there is a need for both variable selection criteria that will enable selection among alternative variable combinations and optimization techniques that can enable the determination of the best variable combination without trying all alternative models according to these variable selection criteria (Bozdogan (2003)).

Model selection procedures based on information criteria are better alternatives to classical hypothesis testing type of approaches. In the literature model selection based on Information Criteria, which is an alternative to classical model selection, is Akaike Information Criterion (AIC), which was first introduced by Akaike (1973). Following Akaike's related work, many criteria have been developed based on the AIC, thus termed as Akaike-type information criteria. The Takeuchi Information Criterion (TIC), also known as the Generalized Akaike Information Criterion (GAIC), proposed by Takeuchi (1976), is based on estimating the Kullback-Leibler (KL) distance between a candidate model and the true model. Bayesian Information Criterion (BIC) developed by Schwarz (1978), Hannan-Quinn Information Criterion (HQC) developed by Hannan & Quinn (1979), the Consistent Akaike Information Criteria (CAIC) and Consistent Akaike Information Criterion (CAICF) which proposes to use the Fisher information matrix, developed by Bozdogan (1987) are some of these criteria. Liu (2007) proved that these criteria yield better results than the classical approaches that have been frequently used so far. Selecting the best model is an important problem when there is no precise information about the relationships between variables. From this point of view, Bozdogan (1988, 1990a,b) developed ICOMP information criteria by considering not only the goodness of fit

and model simplicity, but also the complexity of the model. ICOMP is based on the generalization and utilization of van Endem (1971) covariance complexity measure in multivariate linear model estimation. ICOMP uses an information-based characterization of the covariance matrix of a model's parameter estimates starting from their finite sampling distributions and the complexity of the inverse-Fisher information matrix in linear or nonlinear statistical models.

Information criteria are closely associated with the Kullback-Leibler (KL) distance, as they help measure and minimize the difference between an estimated model and the true data-generating process. The KL distance is a key concept in information theory and is used in many criteria, such as AIC, to balance the model fit and complexity. To better understand the connection between information criteria and KL distance, it is important to explain key ideas like entropy, the Fisher Information Matrix (FIM), and the KL distance that form the basis of these model selection criteria.

2.1 Kullback-Liebler Distance and Fisher Information

Let f and g represent two probability functions corresponding to two statistical models, where f denotes the true model and g the estimated model. The KL distance from f to g is given by

$$I(f, g) = \int f(x) \log\left(\frac{f(x)}{g(x|\theta)}\right) dx, \quad (2.1)$$

where x is the data, and θ is the model parameter vector. The KL distance is often represented by $I(f, g)$ since approximation f results in a loss of information. The above equation can also be expressed as

$$I(f, g) = \int f(x) \log(f(x)) dx - \int f(x) \log(g(x|\theta)) dx \quad (2.2)$$

and it is equivalent to

$$I(f, g) = E_f[\log(f(x))] - E_f[\log(g(x|\theta))]. \quad (2.3)$$

is calculated. Let $X = (x_1, x_2, \dots, x_n)$ be a random variable with n independent observations from a probability density function $f(X|\theta)$ the likelihood function is defined by

$$L(\theta|X) = \prod_{i=1}^n f_i(x_i|\theta), \quad (2.4)$$

and the log-likelihood function is defined

$$l(\theta) = \log(\theta|X) = \sum_{i=1}^n \ln f_i(x_i|\theta). \quad (2.5)$$

Then the expected Fisher information is defined as

$$\mathcal{F}(\theta) = -E \left(\frac{\partial^2 l(\theta)}{\partial \theta^2} \right), \quad (2.6)$$

Equation 2.6 represents a non-negative information matrix closely related to the concepts of efficiency and sufficiency. It measures the amount of information about the parameter θ and is linearly proportional to the unbiased estimator of θ . Fisher information and KL information share similar properties regarding sufficiency, efficiency, additivity, and grouping of observations.

2.2 Akaike Information Criterion (AIC)

AIC (Akaike Information Criterion) is widely used as one of the most common model selection criteria. Because it is based on an unknown truth, KL distance is not enough by itself as a model selection criterion. Akaike (1973) estimated the KL distance between the true and candidate models using likelihood theory. The purpose of Akaike was to minimize this loss as well as also keeping the model as basic as feasible. The bias in model estimation is related to the number of parameters, denoted as k . Therefore, the bias from the estimated expected log-likelihood was subtracted. In order to eliminate this problem, AIC includes a penalty term depending on the number of estimated parameters. This penalty term is found while minimizing the

information loss, KL distance.

$$AIC = -2\log L(\hat{\theta}) + 2k, \quad (2.7)$$

where $L(\hat{\theta})$ is the maximized likelihood function, k is the number of estimated parameters in the model, $\hat{\theta}$ is the Maximum Likelihood Estimate (MLE) of θ . In the AIC formula, the first term represents the lack-of-fit component, while $2k$ acts as the penalty term. The model with the lowest AIC value is selected as the best fit for the data. This approach ensures that the chosen model minimizes information loss, making it the closest approximation to the true model.

2.3 Information Complexity Criterion (ICOMP)

The informational complexity ICOMP criterion is based on the covariance complexity index of van Endem (1971) in parametric estimation. Instead of penalizing the number of free parameters directly, ICOMP penalizes the covariance complexity of the model. For a general multivariate linear or nonlinear model defined by

$$\text{Statistical Model} = \text{Signal} + \text{Noise}. \quad (2.8)$$

ICOMP is designed to estimate a loss function

$$\left. \begin{array}{l} \text{Lack of fit} \\ \text{Loss} = +\text{Lack of Parsimony} \\ +\text{Profusion of Complexity} \end{array} \right\} \Rightarrow AIC \quad \left. \vphantom{\begin{array}{l} \text{Lack of fit} \\ \text{Loss} = +\text{Lack of Parsimony} \\ +\text{Profusion of Complexity} \end{array}} \right\} \Rightarrow ICOMP$$

in several ways using the additivity properties of information theory. In AIC, the compromise takes place between the maximized log-likelihood, $-2\log L(\hat{\theta})$ (the lack of fit component) and k , the number of free parameters estimated within the model (the penalty component) which is a measure of complexity that compensates for the bias in the lack of fit when the MLE are used. On the other hand, ICOMP has a third term in the loss function which is called the 'Profusion of Complexity' which measures how the parameter estimates are correlated with one another in the model

fitting process. Therefore, instead of penalizing the number of free parameters directly, ICOMP penalizes the covariance complexity of the model. It is defined by

$$ICOMP = -2\log L(\hat{\theta}) + 2C(\hat{\Sigma}_{model}) \quad (2.9)$$

where L denotes the likelihood function, and $\hat{\theta}$ is the estimator of the unknown parameter θ , C represents a measure of complexity, and the factor of 2 is included for consistency with the original formulation of AIC, which incorporates a $-2\log L(\hat{\theta})$ lack-of-fit term. Additionally, the model represents the estimated covariance matrix of the parameter vector obtained from the model. Assuming normality, the original covariance complexity, as defined by van Endem (1971), is expressed as:

$$C_0(\Sigma) = \frac{1}{2} \sum_{j=1}^k \log \sigma_{jj}^2 - \frac{1}{2} \log |\Sigma|, \quad (2.10)$$

where k denotes the dimension of Σ , and the σ_{jj}^2 represents the diagonal elements of Σ . Notably, $C_0(\Sigma) = 0$ when Σ is a diagonal matrix. If θ is a normal random vector with covariance matrix Σ , then $C_0(\Sigma) = 0$ corresponds to the Kullback-Leibler divergence between the multivariate normal density of θ and the product of the marginal densities of its components. However, as noted by van Endem (1971), a limitation of C_0 as a complexity measure is its lack of invariance under orthonormal transformations. To address this limitation, Bozdogan (1988, 1990b, 1993, 1994) introduced a maximal information complexity measure based on the C_0 measure, the idea of using a maximal informational complexity measure denoted by C_1 ,

$$C_1(\Sigma) = \max_T C_0(\Sigma) = \frac{k}{2} \log \frac{\text{tr}(\Sigma)}{k} - \frac{1}{2} \log |\Sigma| \quad (2.11)$$

where T represents the set of all orthonormal transformations, and k is the dimension of Σ .

2.3.1 ICOMP (IFIM)

The most general form of ICOMP, called ICOMP (IFIM), takes advantage of the well-known asymptotic optimality properties of MLEs and uses the Inverse Fisher Matrix (IFIM) to measure the complexity of a model. In this case, the most general form of ICOMP is given by

$$ICOMP(IFIM) = -2\log L(\hat{\theta}) + 2C_1(\hat{\mathcal{F}}^{-1}) \quad (2.12)$$

where $C_1(\hat{\mathcal{F}}^{-1})$ indicates the maximal information complexity of the estimated Fisher information matrix with s is the $s = \dim(\hat{\mathcal{F}}^{-1}) = \text{rank}(\hat{\mathcal{F}}^{-1})$ Bozdogan (1990b, 2000, 2003).

$$C_1(\hat{\mathcal{F}}^{-1}) = \frac{s}{2} \log L \left(\frac{\text{tr}(\hat{\mathcal{F}}^{-1})}{s} \right) - \frac{1}{2} \log |\hat{\mathcal{F}}^{-1}| \quad (2.13)$$

In the multivariate case, especially in multivariate Gaussian case, the Inverse Fisher Information Matrix (IFIM) is closely tied to the model covariance matrix and plays a crucial role in building information complexity. It is define by

$$\mathcal{F}^{-1} = \begin{bmatrix} \hat{\Sigma}(\hat{\theta}) & 0 \\ 0 & \frac{2}{n} D_p^+ \left[\hat{\Sigma}(\hat{\theta}) \otimes \hat{\Sigma}(\hat{\theta}) \right] D_p^{+'} \end{bmatrix}, \quad (2.14)$$

where D_p is the duplication matrix of size $p^2 \times p(p+1)/2$. The Moore-Penrose inverse of D_p , denoted as $D_p^+ = (D_p' D_p)^{-1} D_p'$ and it is a matrix of size $p(p+1)/2 \times p^2$. The symbol $\otimes$ denotes the Kronecker product (Williams et al. (1995)).

CHAPTER THREE

A NEWLY PROPOSED ICOMP-ROC CRITERION

3.1 Universal ROC Curve

Consider a pair of random variables, X and Y with the joint distribution $\mathbb{Q}$. In this context, X represents a real-valued score, while Y is a binary event, with the implicit understanding that higher values of X indicate stronger support for the event Y to occur ($Y = 1$). The joint distribution $\mathbb{Q}$ for the pair (X, Y) is defined by the prevalence that falls within the range of $[0, 1]$. Let $\pi_0 = 1 - \pi_1 = \mathbb{Q}(Y = 0)$, and let

$$F(x) = \mathbb{Q}(X \leq x) = \pi_0 F_0(x) + \pi_1 F_1(x) \quad (3.1)$$

denote the marginal cumulative distribution function (cdf) of the score X . In addition, this distribution is characterized by conditional cdf defined by

$$F_1(x) = \mathbb{Q}(X \leq x | Y = 1) \quad \text{and} \quad F_0(x) = \mathbb{Q}(X \leq x | Y = 0). \quad (3.2)$$

Any threshold value x can be used to predict a positive outcome ($Y = 1$) if $X > x$ and a negative outcome ($Y = 0$) if $X \leq x$, to yield a classifier with TPR and FPR, as given in Equations 3.3 and 3.4, respectively.

$$TPR(x) = \mathbb{Q}(X > x | Y = 1) = 1 - F_1(x) \quad (3.3)$$

$$FPR(x) = \mathbb{Q}(X > x | Y = 0) = 1 - F_0(x) \quad (3.4)$$

The ROC curve is a representation that is created through the linear interpolation of raw ROC diagnostics. It is also a point set that may admit a direct interpretation as a function. In the case where both F_1 and F_0 are continuous and strictly increasing functions, the raw ROC diagnostic and the ROC curve can be identified by a function

R, as given in Equation 3.5

$$R(p) = \begin{cases} 0, & p = 0, \\ 1 - F_1(F_0^{-1}(1 - p)), & p \in (0, 1), \\ 1, & p = 1. \end{cases} \quad (3.5)$$

In data analytic practice, the measure $\mathbb{Q}$ is the empirical distribution of a sample $(x_i, y_i)_{i=1}^n$ of real-valued scores x_i and corresponding binary observations y_i . Considering the unique values of $x_1, \dots, x_n$ is sufficient for generating the raw ROC diagnostic, and linear interpolation yields the empirical ROC curve (Gneiting & Walz (2022)).

3.2 A Newly Proposed ICOMP-ROC Criterion

To define ICOMP-ROC criterion, let X_1 and X_2 denote *False Positive Rate (FPR)* and *True Positive Rate (TPR)*. Let (X_1, X_2) have the bivariate normal (Gaussian) distribution with the joint probability density function given by

$$\begin{aligned} f_{X_1 X_2}(x_1, x_2) &= \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \\ &\times \exp \left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x_1 - \mu_1}{\sigma_1} \right)^2 \right. \right. \\ &\quad \left. \left. - 2\rho \left(\frac{x_1 - \mu_1}{\sigma_1} \right) \left(\frac{x_2 - \mu_2}{\sigma_2} \right) \right. \right. \\ &\quad \left. \left. + \left(\frac{x_2 - \mu_2}{\sigma_2} \right)^2 \right] \right\} \end{aligned} \quad (3.6)$$

where ρ is the correlation coefficient between X_1 and X_2 . It is given by

$$\rho = \frac{Cov(x_1, x_2)}{\sigma_1\sigma_2}. \quad (3.7)$$

In matrix notation, we denote the bivariate normal distribution as

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N_{p=2} \left(\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \right). \quad (3.8)$$

This pdf is often used in many applications to model the joint pdf of two random variables X_1 and X_2 . It has five parameters $\theta = \{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho\}$.

The *standard bivariate normal distribution* in terms of the *sample correlation matrix* R is denoted as

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N_{p=2} \left(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix} \right). \quad (3.9)$$

In general, the probability density function (pdf) of the p -dimensional multivariate normal (or Gaussian) distribution is

$$f(x, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp \left(-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)' \right), \quad (3.10)$$

where x and μ are $(1 \times p)$ vectors and Σ is a $(p \times p)$ symmetric, positive definite matrix.

Under the bivariate normal, in terms of the correlation matrix R , the *estimated inverse Fisher information matrix (IFIM)* after some work is given by

$$\hat{\mathcal{F}}^{-1} = \begin{bmatrix} R & 0 \\ 0 & \frac{2}{n}D_2^+(R \otimes R)D_2^{+'} \end{bmatrix} \quad (3.11)$$

where $D_p^+ = (D_p' D_p)^{-1} D_p'$ is the Moore-Penrose inverse of the duplication matrix. The duplication matrix is a unique $p^2 \times \frac{1}{2}p(p+1)$ matrix which transforms, for symmetric matrix A , $vech(A)$ into $vec(A)$. That is,

$$D_p vech(A) = vec(A) \quad (A = A'), \quad (3.12)$$

where $vech(A)$ denotes $\frac{1}{2}p(p+1) \times 1$ vector that is obtained from $vec(A)$ by

eliminating all subradiagonal elements of A . For example, for $p = 2$, that is, for a (2×2) symmetric matrix A , we have

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad \text{vec}(A) = (a_{11}, a_{21}, a_{12}, a_{22})' \quad \text{and} \quad \text{vech}(A) = (a_{11}, a_{21}, a_{22})' \quad (3.13)$$

where the supradiagonal element a_{12} has been removed. Then

$$D_2 \text{vech}(A) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{21} \\ a_{22} \end{pmatrix} = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{21} \\ a_{22} \end{pmatrix} = \text{vec}(A). \quad (3.14)$$

Duplication matrix improves the computational time of very large estimated IFIM in many applications.

Using the definition of ICOMP in Equation 2.12, based on *IFIM*, we now define the ICOMP-ROC criterion as

$$ICOMP - ROC(\hat{\mathcal{F}}^{-1}) = -2\log L(\hat{\theta}) + 2C_1(\hat{\mathcal{F}}^{-1}), \quad (3.15)$$

and derive its analytical form for the ROC Curve model as our fitness function given by

$$ICOMP - ROC(\hat{\mathcal{F}}^{-1}) = n(\log(2\pi) + \log|R| + 1) + 2C_1(\hat{\mathcal{F}}^{-1}), \quad (3.16)$$

where $n = n_1 + n_2$ is the total sample size. Note that the sample sizes do not need to be equal. Further

$$C_1(\hat{\mathcal{F}}^{-1}) = \frac{s}{2} \log \left(\frac{\text{tr}(\hat{\mathcal{F}}^{-1})}{s} \right) - \frac{1}{2} \log |\hat{\mathcal{F}}^{-1}| \quad (3.17)$$

is the maximal entropic complexity of $\hat{\mathcal{F}}^{-1}$, *IFIM* of the bivariate normal (Gaussian) model.

For the bivariate case, for $p = 2$, the opened up form of $C_1(\hat{\mathcal{F}}^{-1})$ in terms of the correlation matrix R is obtained as

$$C_1(\hat{\mathcal{F}}^{-1}) = \frac{s}{2} \log \left(\frac{\text{tr}(R) + \frac{1}{n} [\text{tr}(R) + (\text{tr}(R))^2 + 4]}{s} \right) - 2 \log |R| + \frac{3}{2} \log(n) - \log(2), \quad (3.18)$$

where $s = \text{rank}(\hat{\mathcal{F}}^{-1})$. Note that by computing the complexity this way, we avoid building the large IFIM, and we only need traces and determinants of IFIM, which is computationally efficient. AIC-ROC is defined by

$$AIC - ROC = n(\log(2\pi) + \log|R| + 1) + 2\text{rank}(R). \quad (3.19)$$

The rationale for proposing bivariate normal (Gaussian) distribution to fit and score the ROC curve models stems from the fact that FPR and TPR are dependent, rather than being independent.

Similar to the interpretation of AIC and ICOMP values, the best result is determined by the minimum value of AIC-ROC and ICOMP-ROC. Lower values of these criteria indicate a better model fit and a more accurate identification of the true distribution. This follows the general principle in model selection, where minimizing AIC or ICOMP reflects an optimal balance between model complexity and goodness of fit, as commonly discussed in the literature Bozdogan (1987, 2000). Applying this concept to ROC analysis ensures that the models with the lowest AIC-ROC and ICOMP-ROC values are considered the most suitable.

3.3 Large Scale Monte Carlo Simulation Studies

In this section, we present the large-scale Monte Carlo simulations that empirically compare the AIC-ROC and ICOMP-ROC criteria with the widely known AUC from the literature. The aim of these simulations is to demonstrate that when the distributions of the negative class (X_1) and positive class (X_2) are unknown,

the ICOMP-ROC criterion is more effective at identifying the correct distribution compared to AUC. For this purpose, we assume that the distributions of X_1 and X_2 are derived from Normal–Normal and Weibull–Gamma distributions, respectively, with sample sizes of $n_{X_1} = 200$ and $n_{X_2} = 300$. Then, sixteen different Bi-distribution combination scenarios are constructed by using the probability distributions Normal, Exponential, Weibull, Gamma, Extreme Value (EV), Generalized Extreme Value (GEV), Generalized Pareto (GP) for the negative class X_1 and positive class X_2 . Table 3.1 represents several of these distributions. Table 3.2 presents the cumulative distribution functions (cdf), and Table 3.3 presents the inverse cumulative distributions (icdf) of the probability distributions in Table 3.1.

Table 3.1 Probability distributions

Distributions	pdf
Normal	$f(x \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, \text{ for } x \in \mathbb{R}$
Exponential	$f(x \mu) = \frac{1}{\mu} e^{-\frac{x}{\mu}}$
Weibull	$f(x \alpha, \beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^\alpha}, x > 0, \alpha, \beta > 0$
Gamma	$f(x \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, x > 0, \alpha, \beta > 0$
EV	$f(x \mu, \sigma) = \left(\frac{1}{\sigma}\right) e^{\left(\frac{x-\mu}{\sigma}\right)} \exp \left(-e^{\left(\frac{x-\mu}{\sigma}\right)} \right)$
GEV	$f(x \mu, \sigma, k) = \left(\frac{1}{\sigma}\right) g(x) e^{-g(x)}$
	where $g(x) = \left[1 + k \left(\frac{x-\mu}{\sigma}\right)\right]^{-1/k}$ if $k \neq 0$
GP	$f(x \mu, \sigma, k) = \left(\frac{1}{\sigma}\right) \left(1 + \frac{k(x-\mu)}{\sigma}\right)^{\left(-\frac{1}{k}-1\right)}$
	for $x \geq \mu$ when $k \geq 0$, and $\mu \leq x \leq \mu - \sigma/k$ when $k < 0$

Table 3.2 Cumulative distribution functions

Distributions	cdf
Normal	$F(x \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\left(\frac{t-\mu}{\sigma}\right)^2} dt$
Exponential	$F(x \mu) = 1 - e^{-\frac{x}{\mu}}$
Weibull	$F(x \alpha, \beta) = 1 - e^{-\left(\frac{x}{\beta}\right)^\alpha}$
Gamma	$F(x \alpha, \beta) = \frac{I(\alpha, \beta x)}{\Gamma(\alpha)}$
	$I(\alpha, \beta x) = \text{incomplete gamma function}$
EV	$F(x \mu, \sigma) = 1 - \exp\left(-e^{\left(\frac{x-\mu}{\sigma}\right)}\right)$
GEV	$F(x \mu, \sigma, k) = e^{-g(x)}$
GP	$F(x \mu, \sigma, k) = 1 - \left(1 + \frac{k(x - \mu)}{\sigma}\right)^{-1/k}$ for $k \neq 0$

Table 3.3 Inverse cumulative distribution functions

Distributions	icdf
Normal	$x_p = F^{-1}(p \mu, \sigma) = \{x : F(x \mu, \sigma) = p\}$
	where $p = F(x \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\left(\frac{t-\mu}{\sigma}\right)^2} dt$
Exponential	$x_p = F^{-1}(p \mu) = -\mu \log(1 - p)$
Weibull	$x_p = F^{-1}(p \alpha, \beta) = -\alpha [\log(1 - p)]^{1/\beta}$
Gamma	$x_p = F^{-1}(p \alpha, \beta) = \{x : F(x \alpha, \beta) = p\}$
	where $p = F(x \alpha, \beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} \int_0^x t^{\alpha-1} e^{-\frac{t}{\beta}} dt$
EV	$x_p = F^{-1}(p \mu, \sigma)$
GEV	$x_p = F^{-1}(p \mu, \sigma, k) = \mu + \frac{\sigma}{k} ((-\log(p))^{-k} - 1)$ for $k > 0$
GP	$x_p = F^{-1}(p \mu, \sigma, k)$

These cdfs and icdfs are then used to derive the Universal ROC Curve (UROC) in Equation 3.5 by pairing combination of each of these distributions. For illustration, the Weibull-Gamma pair as our pseudo-true distribution are obtained by Equation 3.20 for Monte Carlo Simulation.

$$R_{WG}(p) = 1 - F_W \left(\left(F_G^{-1} \left(1 - p, \hat{\alpha}_G, \hat{\beta}_G \right) \right), \hat{\alpha}_W, \hat{\beta}_W \right), \quad (3.20)$$

where F_W is the cumulative distribution function (cdf) of the Weibull probability density function (pdf) and F_G^{-1} is the inverse cumulative distribution function (icdf)

of the Gamma pdf.

Let $(\hat{\alpha}_W, \hat{\beta}_W)$ and $(\hat{\alpha}_G, \hat{\beta}_G)$ denote the maximum likelihood estimators (MLEs) of the Weibull and Gamma distribution, respectively. The MLEs of β_W and α_W for the Weibull distribution are the solutions of the simultaneous equation:

$$\hat{\beta}_W = \left[\frac{1}{n} \sum_{i=1}^n x_i^{\hat{\alpha}_W} \right]^{1/\hat{\alpha}_W} \quad (3.21)$$

$$\hat{\alpha}_W = \frac{n}{n \log \hat{\beta}_W + \left(\frac{1}{\hat{\beta}_W} \right)^{\hat{\alpha}_W} \sum_{i=1}^n x_i^{\hat{\alpha}_W} \log \left(\frac{x_i}{\hat{\beta}_W} \right) - \sum_{i=1}^n \log x_i}. \quad (3.22)$$

Similarly, for the Gamma distribution, using the log-likelihood function

$$\log L(\alpha_G, \beta_G | x) = n(\alpha_G \log \beta_G - \log \Gamma(\alpha_G)) - (\alpha_G - 1) \sum_{i=1}^n \log x_i - \beta_G \sum_{i=1}^n x_i \quad (3.23)$$

we obtain the MLEs by solving the following equations:

$$\frac{\partial \log L}{\partial \alpha_G} = n \left(\log \beta_G - \frac{d}{d\alpha_G} \log \Gamma(\alpha_G) \right) + \sum_{i=1}^n \log x_i = 0 \quad (3.24)$$

$$\frac{\partial \log L}{\partial \beta_G} = n \frac{\alpha_G}{\beta_G} - \sum_{i=1}^n x_i = 0. \quad (3.25)$$

From the second equation, we obtain $\bar{x} = \frac{\alpha_G}{\beta_G}$. Substituting $\beta_G = \frac{\alpha_G}{\bar{x}}$ into the first equation, we have

$$n \left(\alpha_G - \log \bar{x} - \frac{d}{d\alpha_G} \log \Gamma(\alpha_G) + \log \bar{x} \right) = 0 \quad (3.26)$$

where $\frac{d}{d\alpha_G} \log \Gamma(\alpha_G) = \psi(\alpha_G)$ is known as the digamma function. Using Equation 3.26, we obtain the MLE of α_G .

From the above equations, we note that for the Weibull-Gamma pair, the maximum likelihood estimators (MLEs) cannot be obtained in a closed analytical form. In such cases, to find the MLEs we use numerical optimization algorithms. These algorithms begin by assuming starting initial values for the unknown parameters and then proceed iteratively until a convergence or stopping criterion is satisfied.

The estimation of the Area Under the Curve (AUC) is inherently dependent upon the estimation of the ROC curve. The AUC is

$$AUC = \int_0^1 R_{WG}(p) dp \quad (3.27)$$

$$= \int_0^1 1 - F_W \left(\left(F_G^{-1} \left(1 - p, \hat{\alpha}_G, \hat{\beta}_G \right) \right), \hat{\alpha}_W, \hat{\beta}_W \right) dp. \quad (3.28)$$

In general, for Bi-distributions this integral numerically is evaluated using the Gauss-Kronrod Quadrature (GKQ) algorithm. See, Calvetti et al. (2000).

In the final step of the simulation, ICOMP-ROC and AIC-ROC values are computed for the Weibull and Gamma distributions by using Equations 3.15 and 3.19, respectively. These steps are then repeated for each of the sixteen Bi-distribution pairs, with the criteria recalculated for every pair.

3.3.1 Results of Monte Carlo Simulation Study

3.3.1.1 Case 1: Weibull-Gamma Bi-Distribution Pair

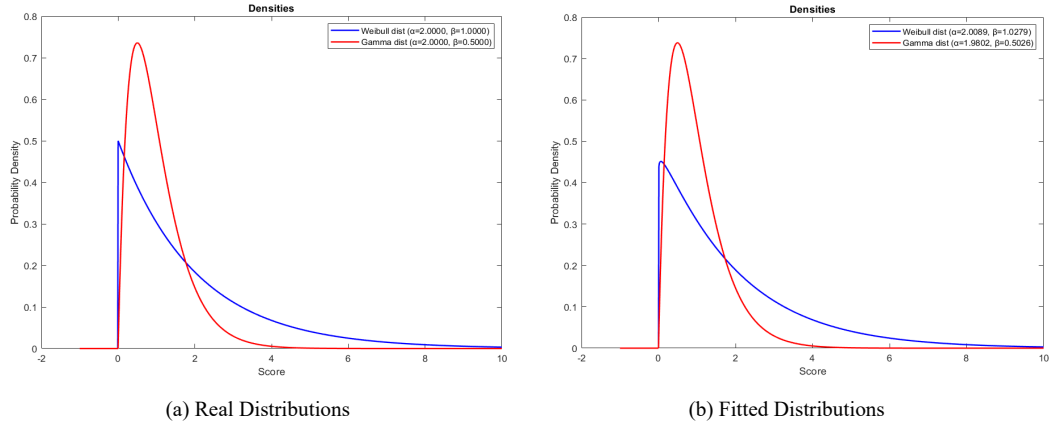


Figure 3.1 Demonstration of real (a) and fitted (b) negative and positive class for Weibull-Gamma Bi-distribution pair

Figure 3.1a and Figure 3.1b represent the real and fitted distribution of Weibull-Gamma pair as our pseudo-true distribution, respectively.

In the simulation study for pseudo-true Weibull-Gamma distribution pair, Table 3.4

displays the best AUC value is obtained for the Weibull-Exponential distribution, while the lowest ICOMP-ROC and AIC-ROC values are achieved for the Weibull-Gamma distribution. As a result, the ICOMP-ROC and AIC-ROC criteria correctly identify this distribution. In contrast, the AUC incorrectly predicts the distribution. Moreover, it is essential to highlight that the AUC, lacks the information criteria foundations provided by as AIC-ROC and ICOMP-ROC.

Table 3.4 Results obtained by running the Monte Carlo simulation once for the Weibull-Gamma Bi-distribution pair

Positive Class	Negative Class	AUC	AIC-ROC	ICOMP-ROC
Normal(1.98,1.87)	Normal(0.99, 0.71)	0.3106	1003.7150	1001.4560
Exponential(1.98)	Exponential(0.99)	0.3338	953.0782	951.8567
Weibull(2.00, 1.02)	Weibull(1.10, 1.47)	0.3542	957.7638	956.3931
Gamma(1.03, 1.91)	Gamma(1.98, 0.50)	0.3542	960.5215	959.0718
EV(3.03, 2.54)	EV(1.39, 0.93)	0.3620	988.2994	986.2680
GEV(0.49, 0.92, 0.90)	GEV(0.16, 0.46, 0.64)	0.3579	955.4247	954.1259
GP(-0.07, 2.13)	GP(-0.25, 1.23)	0.3340	952.7373	951.5275
Exponential(1.98)	Normal(0.99, 0.71)	0.3693	951.8894	950.7093
Gamma(1.03, 1.91)	Normal(0.99, 0.71)	0.3641	952.3061	951.1113
Weibull(2.00, 1.02)	Normal(0.99, 0.71)	0.3624	952.7872	951.5757
EV(3.03, 2.54)	Normal(0.99, 0.71)	0.3676	1006.6130	1004.3160
GEV(0.49, 0.92, 0.90)	Normal(0.99, 0.71)	0.3774	941.7358	940.9978
GP(-0.07, 2.13)	Normal(0.99, 0.71)	0.3571	955.0989	953.8105
Weibull(2.00, 1.02)	Gamma(1.98, 0.50)	0.6475	938.0450	937.5293
Weibull(2.00, 1.02)	Exponential(0.99)	0.6728	944.7688	943.8782
Gamma(1.03, 1.91)	Exponential(0.99)	0.6715	944.0736	943.2161

Note: Normal(μ, σ) with mean μ and standard deviation σ ; Exponential(λ) with rate parameter λ ; Weibull(α, β) with scale β , and shape α ; Gamma(α, β) with scale β , and shape α ; Extreme Value (EV)(μ, σ) with mean μ and standard deviation σ ; Generalized Extreme Value (GEV) (k, σ, μ) with location parameter μ , scale parameter σ , and shape parameter k ; Generalized Pareto (GP)(k, σ) with shape parameter k and, scale parameter σ .

According to Figure 3.2, the Weibull-Exponential and Gamma-Exponential distribution pairs are closest to the top-left corner and exhibit very close ROC curves. However, the real distribution pair could not be reliably identified.

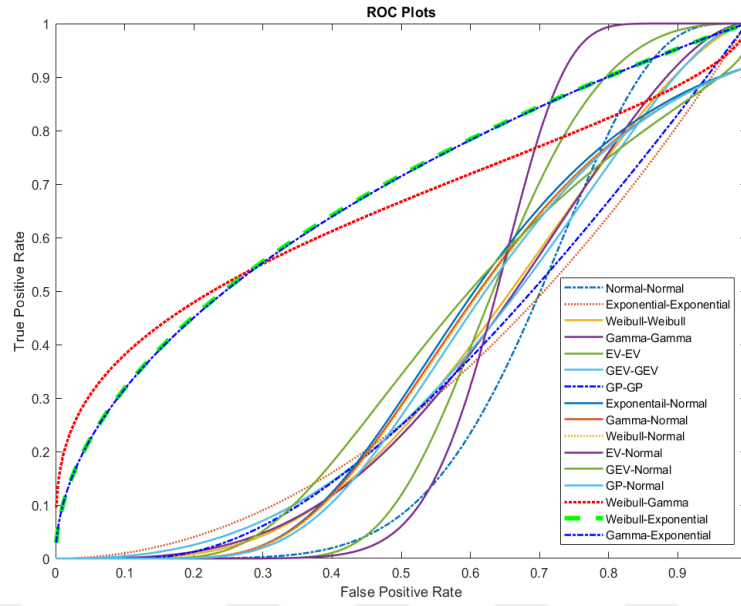


Figure 3.2 The ROC curves for the sixteen Bi-distribution pairs for pseudo-true Weibull-Gamma Bi-distribution pair

Table 3.5 illustrates the frequency of success for the Monte Carlo simulation study across different distribution pairs, comparing the performance of AUC, AIC-ROC, and ICOMP-ROC for 100 runs. The results indicate that the Weibull-Gamma distribution pair achieves a 100% success rate using both AIC-ROC and ICOMP-ROC, correctly identifying the true distribution. In contrast, AUC identifies the Weibull-Exponential distribution as the best fit, with a 100% success rate in this scenario, but fails to recognize the correct distribution for Weibull-Gamma. As a result, this further supports the notion that the AIC-ROC and ICOMP-ROC criteria are more reliable and consistent in identifying the real distribution pair, particularly when accounting for information criteria.

Table 3.5 Frequency of success for the Monte Carlo simulation study for the Weibull-Gamma Bi-distribution pair

Positive Class	Negative Class	AUC	AIC-ROC	ICOMP-ROC
Normal(1.98,1.87)	Normal(0.99, 0.71)	0	0	0
Exponential(1.98)	Exponential(0.99)	0	0	0
Weibull(2.00, 1.02)	Weibull(1.10, 1.47)	0	0	0
Gamma(1.03, 1.91)	Gamma(1.98, 0.50)	0	0	0
EV(3.03, 2.54)	EV(1.39, 0.93)	0	0	0
GEV(0.49, 0.92, 0.90)	GEV(0.16, 0.46, 0.64)	0	0	0
GP(-0.07, 2.13)	GP(-0.25, 1.23)	0	0	0
Exponential(1.98)	Normal(0.99, 0.71)	0	0	0
Gamma(1.03, 1.91)	Normal(0.99, 0.71)	0	0	0
Weibull(2.00, 1.02)	Normal(0.99, 0.71)	0	0	0
EV(3.03, 2.54)	Normal(0.99, 0.71)	0	0	0
GEV(0.49, 0.92, 0.90)	Normal(0.99, 0.71)	0	0	0
GP(-0.07, 2.13)	Normal(0.99, 0.71)	0	0	0
Weibull(2.00, 1.02)	Gamma(1.98, 0.50)	0	100	100
Weibull(2.00, 1.02)	Exponential(0.99)	100	0	0
Gamma(1.03, 1.91)	Exponential(0.99)	0	0	0

Note: Normal(μ, σ) with mean μ and standard deviation σ ; Exponential(λ) with rate parameter λ ; Weibull(α, β) with scale β , and shape α ; Gamma(α, β) with scale β , and shape α ; EV(μ, σ) with mean μ and standard deviation σ ; GEV(k, σ, μ) with location parameter μ , scale parameter σ , and shape parameter k ; GP(k, σ) with shape parameter k and, scale parameter σ .

3.3.1.2 Case 2: Normal-Normal Bi-Distribution Pair

Figure 3.3a and Figure 3.3b represent the real and fitted distribution of Normal-Normal pair as our pseudo-true distribution, respectively.

In the simulation study for pseudo-true Normal-Normal distribution pair, Table 3.6 displays the AUC, AIC-ROC, and ICOMP-ROC values of sixteen different Bi-distribution combination scenarios, covering both symmetric and asymmetric cases commonly encountered in real-world data. In Table 3.6, the highest AUC value is obtained for the GP-Normal distribution pair, while the lowest AIC-ROC value is

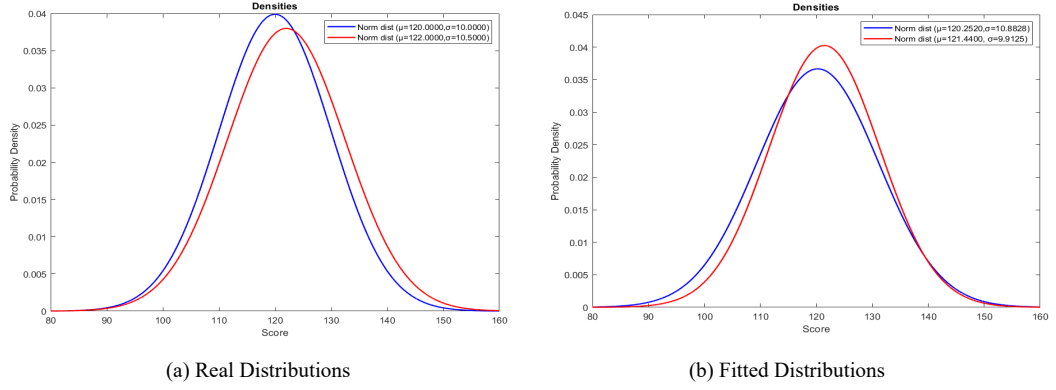


Figure 3.3 Demonstration of real (a) and fitted (b) negative and positive class for the Normal-Normal Bi-distribution pair

observed for the Exponential-Exponential distribution pair, and the lowest ICOMP-ROC value is achieved for the Normal-Normal distribution pair. As a result, the ICOMP-ROC criterion accurately identifies the correct distribution. In contrast, both AIC-ROC and AUC misidentify the distribution. Furthermore, it is important to emphasize that AUC lacks the information-theoretic foundations provided by AIC-ROC and ICOMP-ROC, and that AIC-ROC itself does not account for model complexity as robustly as ICOMP-ROC.

Table 3.6 Results obtained by running the Monte Carlo simulation once for the Normal-Normal Bi-distribution pair

Positive Class	Negative Class	AUC	AIC-ROC	ICOMP-ROC
Normal(120.25,10.88)	Normal(121.44,9.91)	0.5322	924.2721	926.1975
Exponential(120.25,0)	Exponential(121.44,0)	0.5025	922.9439	930.3891
Weibull(125.28,11.03)	Weibull(125.94,13.37)	0.5289	925.1229	926.5540
Gamma(123.19,0.98)	Gamma(147.58,0.82)	0.5335	924.3183	926.2096
EV(125.81,11.71)	EV(126.31,9.47)	0.5261	925.2603	926.6303
GEV(116.01,10.43)	GEV(118.06,10.18)	0.5404	926.7968	927.6573
GP(-1.43,222.86)	GP(-1.6,236.08)	0.4863	925.4887	926.7646
Exponential(120.25,0)	Normal(121.44,9.91)	0.6345	1036.5695	1033.9353
Gamma(123.19,0.98)	Normal(121.44,9.91)	0.5370	925.3379	926.6749
Weibull(125.28,11.03)	Normal(121.44,9.91)	0.5210	925.5925	926.8284
EV(125.81,11.71)	Normal(121.44,9.91)	0.5177	927.6919	928.3429
GEV(116.01,10.43)	Normal(121.44,9.91)	0.5385	926.6435	927.5447
GP(-1.43,222.86)	Normal(121.44,9.91)	0.6558	1019.5691	1017.1146
Weibull(125.28,11.03)	Gamma(147.58,0.82)	0.4817	926.0646	927.1363
Exponential(125.28,11.03)	Exponential(121.44,0)	0.6245	1027.2664	1024.7273
Exponential(123.19,0.98)	Exponential(121.44,0)	0.6270	1032.3725	1029.7804

Note: Normal(μ, σ) with mean μ and standard deviation σ ; Exponential(λ) with rate parameter λ ; Weibull(α, β) with scale β , and shape α ; Gamma(α, β) with scale β , and shape α ; Extreme Value (EV)(μ, σ) with mean μ and standard deviation σ ; Generalized Extreme Value (GEV) (k, σ, μ) with location parameter μ , scale parameter σ , and shape parameter k ; Generalized Pareto (GP)(k, σ) with shape parameter k and, scale parameter σ .

The ROC curves for the sixteen Bi-distribution pairs are presented in Figure 3.4. A curve that is closer to the top-left corner represents a better classifier. According to Figure 3.4, the GP-Normal distribution pair is closest to the top-left corner. However, the real distribution pair could not be reliably identified.

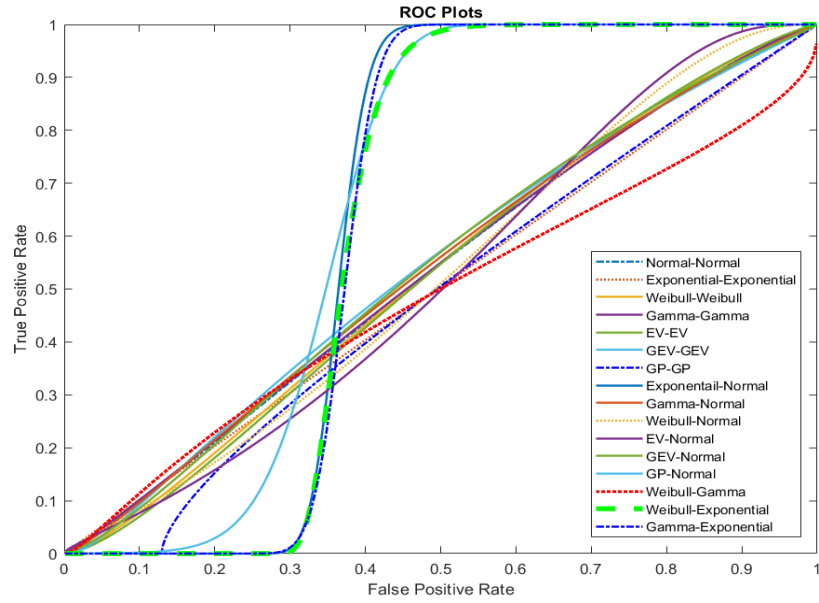


Figure 3.4 The ROC curves for the sixteen Bi-distribution pairs for pseudo-true Normal-Normal Bi-distribution pair

Table 3.7 illustrates the frequency of success for the Monte Carlo simulation study across different distribution pairs, comparing the performance of AUC, AIC-ROC, and ICOMP-ROC for 100 runs. The results indicate that the Normal-Normal Bi-distribution pair achieves a 100% success rate using ICOMP-ROC, correctly identifying the true distribution. In contrast, AUC identifies the GP-Normal distribution pair and AIC-ROC identifies the Exponential-Exponential distribution pair as the best fit, with a 100% success rate in this scenario, but fails to recognize the correct distribution for Normal-Normal. As a result, this further supports the notion that ICOMP-ROC criteria are more reliable and consistent in identifying the real distribution pair, particularly when accounting for information criteria.

Table 3.7 Frequency of success for the Monte Carlo simulation study for Normal-Normal Bi-distribution pair

Positive Class	Negative Class	AUC	AIC-ROC	ICOMP-ROC
Normal(1,1.87)	Normal(0.99, 0.71)	0	0	100
Exponential(1.98)	Exponential(0.99)	0	100	0
Weibull(2.00, 1.02)	Weibull(1.10, 1.47)	0	0	0
Gamma(1.03, 1.91)	Gamma(1.98, 0.50)	0	0	0
EV(3.03, 2.54)	EV(1.39, 0.93)	0	0	0
GEV(0.49, 0.92, 0.90)	GEV(0.16, 0.46, 0.64)	0	0	0
GP(-0.07, 2.13)	GP(-0.25, 1.23)	0	0	0
Exponential(1.98)	Normal(0.99, 0.71)	0	0	0
Gamma(1.03, 1.91)	Normal(0.99, 0.71)	0	0	0
Weibull(2.00, 1.02)	Normal(0.99, 0.71)	0	0	0
EV(3.03, 2.54)	Normal(0.99, 0.71)	0	0	0
GEV(0.49, 0.92, 0.90)	Normal(0.99, 0.71)	0	0	0
GP(-0.07, 2.13)	Normal(0.99, 0.71)	100	0	0
Weibull(2.00, 1.02)	Gamma(1.98, 0.50)	0	0	0
Weibull(2.00, 1.02)	Exponential(0.99)	0	0	0
Gamma(1.03, 1.91)	Exponential(0.99)	0	0	0

Note: Normal(μ , σ) with mean μ and standard deviation σ ; Exponential(λ) with rate parameter λ ; Weibull(α , β) with scale β , and shape α ; Gamma(α , β) with scale β , and shape α ; EV(μ , σ) with mean μ and standard deviation σ ; GEV(k , σ , μ) with location parameter μ , scale parameter σ , and shape parameter k ; GP(k , σ) with shape parameter k and, scale parameter σ .

CHAPTER FOUR

FEATURE SELECTION WITH THE GENETIC ALGORITHM

In the literature, various methods have been employed to improve models, ranging from traditional statistical approaches to more sophisticated machine learning techniques. Linear regression, decision trees, and support vector machines are among the most commonly used methods for optimization and prediction tasks. While these methods perform well on a wide range of problems, they often struggle with high-dimensional, nonlinear, or complex data. To overcome these constraints, advanced optimization techniques such as Genetic Algorithm (GA) are introduced as a novel and superior alternative to conventional approaches.

The GA differs from traditional methods by emulating the process of natural selection. GA is an example of a non-deterministic approach where we are working on a population of possible solutions that are evolved through several generations using operations like selection, crossover, and mutation. This evolutionary mechanism enables GA to cover a larger solution space and escape local optima, which is often advantageous when tackling challenging non-linear problems. GA is employed in this thesis to tackle feature selection problems by finding the best or approximately best subsets of features that enhance the performance of the model while also minimizing its computational complexity.

4.1 Overview of Genetic Algorithm

Computer scientists started investigating an evolving system as a tool for optimization in the 1950s and 1960s. The concept of evolutionary computing was formally introduced by Rechenberg (1973) in his article "Evolution Strategies" (Evolution strategies in the original). The evolutionary approach a large population of alternative answers. Over time, it is permitted for these solutions to interact; the population gets better through random mutations and fitness-based selection, eventually iterating to an ideal solution. The Genetic Algorithm (GA) is a type of

evolutionary algorithm inspired by the principles of natural selection and genetic operators. Initially developed by Holland (1975) along with his colleagues and students, GA has since evolved into a powerful optimization technique. Later, Koza (1994) utilized genetic algorithms to develop programs capable of performing specific tasks. He called his method “genetic programming”. Since then, GA has continued to be widely studied and applied to various fields of science and engineering disciplines. In addition to providing alternative approaches to problem-solving, Genetic Algorithms (GA) often outperform other methods in terms of speed and efficiency when searching highly nonlinear search spaces.

4.2 Basic Terminology

GA starts with a population, which is a set of solutions. An individual in the population is a specific solution. In GA, solutions are represented in the form of a binary string, known as a chromosome, which can take two values: 0 and 1. Inheritance, mutation, and crossover are performed to prevent getting stuck at a local minimum. The key to ranking solutions is termed a fitness function, and the best-fitted are retained to form the next generation. The greater the number of generations, the longer the computation time. However, insufficient iterations can lead to suboptimal results.

The solutions produced during the process are called offspring since GA operates on a population of chromosomes from the current generation to produce pairs of new solutions. When the retained good solutions mate, the new solutions created are also high-quality solutions that we retain. This procedure continues until some convergence condition is satisfied or the optimal solutions are obtained. One commonly used approach is “natural” selection, where the probability of an individual being selected is proportional to its fitness ratio. The fitness ratio, denoted as r_j , is calculated using the formula:

$$r_j = \frac{\Delta Fitness_j}{\overline{\Delta Fitness}}, \quad (4.1)$$

where $\Delta Fitness_j = Fitness_{Max} - Fitness_j$, and $\overline{\Delta Fitness}$ is the mean of $\Delta Fitness_j$, representing the average fitness difference across the population. Typically, the index j represents individuals within the population. When the population size is N , $j = 1, 2, 3, \dots, N$. This ensures that each individual in the population is accounted for when calculating the fitness ratio, which ultimately determines the likelihood of being selected for the next generation.

4.3 GA Operators

The genetic algorithms use three types of operators: selection, crossover, and mutation. Selection is the process of determining which individuals are the fittest within the population, allowing them to pass on their characteristics to the next generation. Crossover splices the genetic information from two parent solutions together to produce one or more offspring, introducing diversity and exploration into the population. Meanwhile, mutation adds small random variations to individual solutions, which helps the algorithm escape from local optima and maintain the population's genetic diversity. Together, these operators enable genetic algorithms to efficiently explore and exploit complex solution spaces. Detailed information of operators is given below.

4.3.1 Selection

Selection is the step where the program picks high-ranking chromosomes for reproduction. Chromosomes are ranked based on a fitness function at each iteration. Chromosomes with higher fitness values have a greater chance of being reproduced compared to those with lower fitness values. Various methods exist for selection operations, most involving some proportionality between fitness and reproduction

likelihood. Roulette wheel sampling, introduced by Goldberg (1989), is a widely used selection method that mimics the assignment of ranked probabilities to proportionate segments of a roulette wheel. Using this method, the relative probabilities of the chromosomes are encoded into a probability distribution, with the highest-ranking probabilities given the greatest weight. This process continues until either a single chromosome dominates the population or no changes occur in the ranks of chromosomes for several iterations. A common approach for tracking population convergence is plotting average fitness and best fitness across iterations. These plots typically resemble exponentially increasing or decreasing functions that asymptotically approach an optimum value.

4.3.2 Crossover

A theory of GA's called the building block hypothesis that substrings of chromosomes that individually contribute to an optimal solution can combine through crossover to construct the optimal string (Mitchell (1998)). The principle behind crossover is that the new chromosome might outperform both parents if it takes the best characteristics from each. Crossover is performed during evolution based on a user-defined probability of crossover, denoted as P_C . This process continues until the same number of individuals as in the previous generation is created. Common methods include single-point crossover, two-point crossover, and uniform crossover.

Single point crossover: This is the simplest method of crossover. A crossover point is selected, and the binary string from the start of the chromosome to the crossover point is copied from the first parent, while the rest is copied from the second parent.

Two point crossover: Two crossover points are selected. The binary string from the beginning of the chromosome to the first crossover point is copied from the first parent, the segment between the first and second crossover points is copied from the

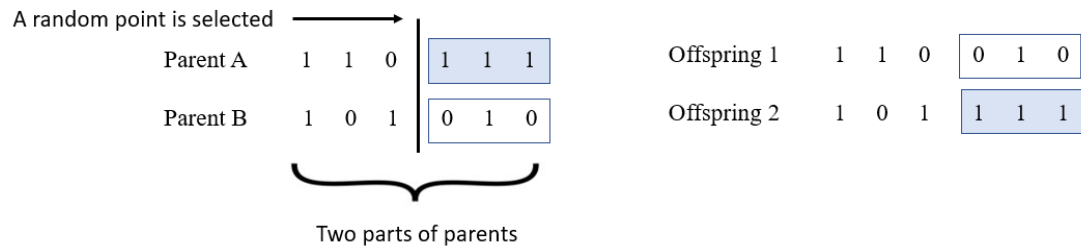


Figure 4.1 Single point crossover

second parent, and the remainder is copied from the first parent.

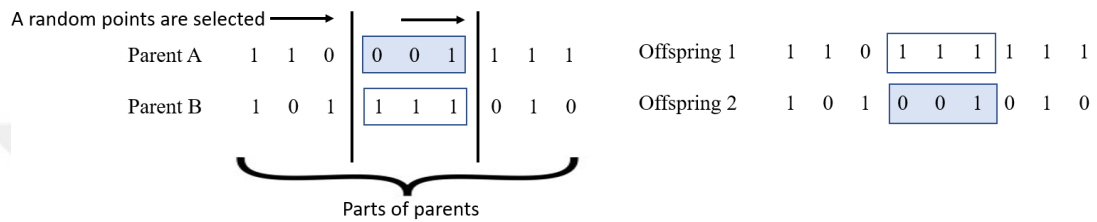


Figure 4.2 Two point crossover

Uniform crossover: In this approach, the bits of the offspring are randomly selected from either the first parent or the second with equal probability. This allows the Genetic Algorithm to better explore the feature space by combining different sets of features from each parent and finding a more diverse set of features (Eiben & Smith (2003)).

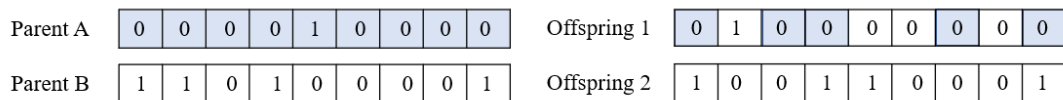


Figure 4.3 Uniform crossover

In summary, the algorithm allows the designs of varied crossover methods, each with its own merits. Single-point crossover provides the advantage of easy and direct combination of traits through splitting at one position, while two-point crossover increases diversity by adding variation at two split points. Uniform crossover is also a possibility, where the bits are selected from either parent with equal probability, making it less likely for the crossover to concentrate in a specific area of the feature space. Additionally, the elitism rule is implemented in the algorithm, ensuring that the best individual in the current population survives.

4.3.3 Mutation

Mutation is a crucial process in GA for introducing diversity. After offspring are generated, mutation occurs. In GA with binary string solutions, this involves randomly selecting solutions and flipping bits ($1 \rightarrow 0$) based on a specified mutation probability, P_M , with a typical mutation rate of less than or equal to 10%. Another approach is to mutate the entire population in one step. For certain problems, intelligently applied biased mutation can be used, modifying solutions based on prior knowledge. Mutation is one of the major advantages of GA. Without it, a solution population might quickly converge and become uniform, getting stuck in a local optimum. The mutation operator, on the other hand, introduces genetic diversity by mutating random bits, widening the search by enabling leaps to other regions of the fitness landscape. Simulated annealing is, to some extent, similar in this respect.

4.3.4 Termination Criteria

Termination methods (in Genetic algorithms) are the methods that decide when to stop the evolution process. Each of the enabled termination criterion is checked after each generation to see if it is time to stop (Mitchell (1996), Whitley (1994)).

One common method is the Generation Number, where evolution ceases after reaching the maximum number of generations specified by the user. The second is Evolution Time, which stops the process once the total elapsed time exceeds a certain threshold, but with the possibility to interrupt mid-generation if required. The Fitness Threshold method ends the process based on whether the best fitness in the population meets (or exceeds) a predefined threshold, based on whether we want to minimize or maximize fitness. Similarly, The Fitness Convergence stops the algorithm whenever it finds that the best fitness value of all generations is stable. Fitness Convergence uses different length filters to measure convergence. In addition, Population Convergence, evolution, stops when the average fitness across the population is close enough to the best fitness value specified by the user. Finally,

Gene Convergence terminates the operation when a certain proportion of the genes stabilizes, obtained by comparing the average gene values in the population with their maximum values. Combined, these methods ensure the algorithm terminates successfully without wasting computation. Parameter configuration of GA is given in Table 4.1

Table 4.1 GA operational parameters

Parameter	Description
N_{Gen}	Number of Generations
N	Population Size
p_C	Probability of Crossover
p_M	Probability of Mutation
Elitism	Keep the Best Subset or Not (On/Off)
Type of Crossover	Single-Point / Two-Point / Uniform
Fitness Function	Information Criteria

The outline of the GA procedures for model parameter estimation and model selection is summarized as follows:

- Step 1: Generate a population of solutions with the specified size N .
- Step 2: Encode the individuals as a binary string (chromosome).
- Step 3: Evaluate and rank all individuals in the population using the fitness function.
- Step 4: Select individuals to be used to generate the new population.
- Step 5: Perform crossover on the selected individuals using the specified crossover type and probability to generate offspring.
- Step 6: Apply mutation to the offspring population based on the given mutation probability.
- Step 7: Optionally, apply genetic engineering techniques to enhance the new population.
- Step 8: Implement elitism, if enabled, to retain the best individual from the current population in the new generation.

Step 9: Replace the current population with the newly generated population.

Step 10: Repeat steps 2 through 9 until a termination condition is met (e.g., maximum generations, fitness threshold, or convergence).



CHAPTER FIVE

OVERVIEW OF MACHINE LEARNING CLASSIFIERS

At the start of the 19th century, Legendre (1805) and Gauss (1809) introduced the method of least squares, the precursor to modern linear regression. Later, Fisher (1936) developed linear discriminant analysis to predict qualitative outcomes, such as patient survival or stock market trends, by identifying linear combinations of features that separate classes. In the 1940s, logistic regression emerged as an alternative, and by the 1970s, Nelder & Wedderburn (1972) introduced the term "generalized linear models" to encompass both linear and logistic regression. By the end of the 1970s, there were many more techniques for learning from data. However, they were mostly linear techniques due to the infeasibility of computing non-linear connections. By the 1980s, advancements in computing technology made it possible to use non-linear methods without facing computational obstacles. The probabilistic approaches emerged thanks to the intersection of computer science and statistics.

Thanks to advances in statistics, machine learning gained significant popularity in the 1990s, leading to the development of intelligent systems capable of analyzing and learning from vast datasets. Since then, inspired by the rise of machine learning and related disciplines, statistical learning has evolved to encompass various techniques for modeling, prediction, and analysis, including supervised, unsupervised, and semi-supervised approaches. Supervised statistical learning, a key subset of machine learning, focuses on training models with labeled datasets to learn mappings between input features and output targets. Traditional methods like linear and logistic regression, alongside modern techniques such as support vector machines, decision trees, random forests, K-nearest neighbors, neural networks, and Naive Bayes, all operate within this framework.

In this chapter, we explore the fundamental concepts and methodologies of widely used supervised machine learning algorithms for binary classification, including Logistic Regression, Support Vector Machine, K-Nearest Neighbor, Naive Bayes, and Decision Tree.

5.1 Logistic Regression Classifier

Logistic regression (LR) is a widely used statistical modeling technique introduced by the statistician Cox (1958). Over the years, LR has become a cornerstone in statistics, machine learning, and data analysis, forming the foundation for many classification and predictive modeling techniques. Logistic regression is designed to predict the probability of a binary outcome, such as "success (1)" or "failure (0)," based on one or more independent variables. Unlike classical regression methods, LR ensures that the predicted values for binary outcomes remain between 0 and 1. This is achieved through the sigmoid function, a mathematical function that maps real-valued inputs to probabilities. The sigmoid function converts the model's linear predictions into probabilities, and a threshold (commonly 0.5) is applied to classify outcomes. Values above the threshold are assigned to one class (e.g., 1), while values below it are assigned to the other class (e.g., 0).

Multiple logistic regression extends this approach by incorporating multiple predictors, allowing for the simultaneous analysis of their combined effects. Each predictor's influence is expressed through coefficients that quantify its impact on the log-odds of the outcome. This makes logistic regression a flexible tool for modeling complex relationships where multiple factors influence the probability of an event.

The performance of logistic regression models is evaluated using metrics such as accuracy, precision, recall, F1-score, ROC, AUC. Additionally, confusion matrices provide detailed insights into classification results by analyzing true positives, true negatives, false positives, and false negatives. Despite its versatility, LR assumes a linear relationship between predictors and the log-odds of the outcome, which can limit its performance in the presence of outliers, imbalanced datasets, or violations of this assumption. In such cases, alternative methods such as Support Vector Machines or Decision Trees may offer better results.

5.2 Support Vector Machine Classifier

Support Vector Machine (SVM) is a supervised learning model in machine learning with associated algorithms that evaluate data for regression and classification tasks. The foundational concept of SVM was introduced by (Vapnik & Lerner (1963)), focusing on linearly separable data. The modern SVM, which incorporates non-linear kernel functions and the soft-margin approach, was developed by Corinna Cortes and Vapnik and formally published in 1995 (Cortes & Vapnik (1995)).

The central idea behind SVM is to identify an optimal hyperplane that maximally discriminates between classes in the training set. This hyperplane is designed to maximize the margin, which is the distance between the closest data points from each class, referred to as support vectors. Maximizing this margin enhances the model's generalizability and improves classification confidence. The dimensionality of the hyperplane corresponds to the number of features in the data: for two features, it is a line; for three features, it becomes a plane. In higher-dimensional spaces, the hyperplane exists as a generalized n -dimensional structure.

SVM supports two types of classification: hard margin and soft margin. Hard margin classification assumes perfectly separable data and seeks a solution with zero classification error, but this approach is highly sensitive to outliers, often leading to overfitting. In contrast, soft margin classification introduces a trade-off by allowing some misclassification or overlap within the margin, balancing the model's robustness and generalizability. This trade-off is controlled by the regularization parameter C , which determines the extent of margin violation allowed.

To handle non-linearly separable data, SVM employs kernel functions, which map input data into higher-dimensional feature spaces, where a linear hyperplane can effectively separate the classes. Commonly used kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. These kernels enable SVM to model complex, non-linear decision boundaries.

Optimization in SVM is achieved through techniques like sequential minimal optimization (SMO), quadratic programming, or iterative single data algorithms. These methods efficiently solve the optimization problem by finding the hyperplane that minimizes the classification error while maximizing the margin. For example, SMO breaks the large optimization problem into smaller sub-problems, significantly improving computation time.

SVM achieves an effective balance between bias and variance by focusing solely on the support vectors, ensuring that the decision boundary is robust and unaffected by data points outside the margin. This balance is further refined by tuning hyperparameters, such as the kernel type and regularization parameter, to achieve optimal performance on unseen data.

In this thesis, SVM is applied to binary classification tasks. The selection of kernel functions and the tuning of regularization parameters are performed to ensure that the model generalizes well to the data at hand. By leveraging the strengths of SVM, the approach combines computational efficiency with high accuracy, particularly for datasets with complex or non-linear structures.

5.3 Naive Bayes Classifier

Naive Bayes (NB) classification, based on Bayes' Theorem developed by Bayes (1763) in the 18th century, represents a group of supervised learning algorithms. NB classifier assumes that the effect of a particular feature in a class is independent of other features. Bayes' Theorem is defined by the following equation:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (5.1)$$

Where:

- $P(A|B)$: Posterior probability of class A given the features B .
- $P(B|A)$: Likelihood of the features B given class A .

- $P(A)$: Prior probability of class A .
- $P(B)$: Evidence or the probability of the features B .

The goal of the Naive Bayes classifier is to assign a class label A to a given observation $P(B)$ by maximizing the posterior probability $P(A|B)$. Since the evidence $P(B)$ is constant for all classes, it is omitted during the maximization process, resulting in the formula:

$$A_{\text{MAP}} = \arg \max_A P(B|A) \cdot P(A) \quad (5.2)$$

Here, A_{MAP} is the class with the highest posterior probability, also known as the Maximum A Posteriori (MAP) estimation. To further simplify the computation, the Naive Bayes algorithm assumes that the features within B are conditionally independent given the class A . This naive assumption allows the likelihood $P(B|A)$ to be expressed as the product of individual feature probabilities:

$$P(B|A) = P(b_1|A) \cdot P(b_2|A) \cdot \dots \cdot P(b_n|A) \quad (5.3)$$

where $B = \{b_1, b_2, \dots, b_p\}$ represents the set of p features. $P(b_i|A)$ is the probability of the i -th feature given class A . This assumption reduces computational complexity, making Naive Bayes highly efficient, especially for high-dimensional data.

Naive Bayes classifiers can be adapted to different types of data distributions. For instance, Gaussian Naive Bayes is used for continuous data and assumes that the data follows a normal distribution. On the other hand, Multinomial Naive Bayes is suitable for discrete data, such as word counts in text classification. At the same time, Bernoulli Naive Bayes is designed for binary data, such as the presence or absence of features in a document.

The Naive Bayes classifier offers notable advantages, including computational efficiency, strong performance with large datasets, robustness despite simplistic assumptions, and ease of implementation and interpretation, making it widely popular across diverse applications. However, the algorithm's reliance on the assumption of

feature independence, imbalanced datasets, and non-conforming continuous variables can limit its accuracy and performance in real-world applications. Despite these challenges, the Naive Bayes is extensively used in fields like spam filtering, medical diagnostics, recommendation systems, and document classification.

5.4 K-Nearest Neighbors Classifier

K-Nearest Neighbors (KNN) is a supervised learning algorithm used for both regression and classification problems, where predictions are made based on the similarity of observations to their nearest neighbors in the feature space. This algorithm was first developed by Fix & Hodges Jr (1951) and later expanded by Thomas Cover (Cover & Hart (1967)). It is often applied as a KNN classifier, where the output is determined by the class membership of the closest neighbors. For classification, the algorithm assigns a new data point to the most common class among its nearest neighbors, with k being a small positive integer.

The KNN algorithm follows a straightforward process. First, a value k , representing the number of neighbors to consider, is selected. For each test data point, the distances to all points in the training set are calculated using a distance metric, such as Euclidean, Manhattan, or Minkowski. The k nearest neighbors are identified, and the majority class among these neighbors determines the class label for the test point. This majority-voting mechanism makes KNN a non-parametric algorithm that does not make assumptions about the underlying data distribution.

The KNN algorithm offers several advantages, including its simplicity, flexibility, and non-parametric nature, which does not require assumptions about the underlying data distribution. However, its performance heavily depends on the appropriate selection of the k value and the choice of a suitable distance metric. In particular, KNN can be computationally expensive and memory-intensive when applied to high-dimensional datasets due to the need to calculate distances for all training points. Additionally, the algorithm tends to be slow for large datasets and sensitive to noisy

data. Therefore, proper data preprocessing, feature selection, and parameter tuning are essential before implementing the KNN algorithm.

5.5 Decision Tree Classifier

Decision Trees (DT) is a non-parametric supervised learning algorithm widely used for regression and classification tasks. They operate by splitting a dataset into smaller subsets based on specific conditions, ultimately forming a tree-like structure where decisions are represented as nodes, and outcomes as leaves. This method is valued for its simplicity, interpretability, and versatility, making it a popular choice in various domains.

The foundation of Decision Tree was laid by Hunt et al. (1966) in the field of computer science and statistics. However, they gained significant traction in the 1980s with the development of the CART (Classification and Regression Trees) algorithm by Breiman (2017). CART formalized the binary tree structure and introduced measures like Gini Impurity and variance reduction for optimal splits.

Another major milestone was the development of the ID3 algorithm by Quinlan (2014), followed by its successors, C4.5 and C5.0, which improved feature selection, pruning mechanisms, and handling of continuous attributes. These advancements made Decision Trees more robust and applicable to complex datasets. In the 1990s, ensemble methods such as Random Forests and Gradient Boosting extended the capabilities of DT by combining multiple trees to enhance performance and reduce overfitting.

The construction of a DT involves the following steps:

Feature Selection for Splitting: At each node, the algorithm selects the feature that best splits the dataset into homogeneous subsets. Metrics like Gini Impurity, information gain (based on entropy), or variance reduction are commonly used to evaluate the quality of a split.

Recursive Partitioning: The selected feature divides the dataset into branches, each corresponding to a unique value or range of the feature. The algorithm then recursively applies the same splitting process to each branch.

Stopping Criteria: The splitting process continues until a predefined stopping condition is met. Common criteria include reaching a maximum tree depth, having a minimum number of samples in a node, or when no further splits improve the model's performance.

Prediction: Once the tree is constructed, predictions for new data points are made by traversing the tree from the root node to a leaf node based on the values of the features.

DT are non-parametric models, meaning they make no assumptions about the data distribution. Although they are powerful tools, they are prone to overfitting, especially with complex or noisy data. To overcome this, techniques such as pruning, regularisation or the incorporation of ensemble methods are often used. The details can be found in James et al. (2013).

CHAPTER SIX

APPLICATION OF ICOMP-ROC TO DENTAL IMPLANT DATA CLASSIFICATION

Many people experience tooth loss due to tooth decay, periodontal diseases, and trauma. In recent years, dental implants have become a popular treatment approach for tooth loss. Dental implants are materials surgically placed into the bone to restore missing teeth with prostheses as shown in Figure 6.1. Dental implants are a highly reliable treatment method with a high success rate in dentistry and are utilized in a wide range of cases, from single-tooth loss to complete edentulism. Numerous clinical studies have demonstrated that implant systems remain functional and successful over the long term (Chochlidakis et al. (2020), Moraschini et al. (2015), Papaspyridakos et al. (2020)). With the developing technology, more than 2000 different types of dental implant systems have been produced, differing in physical and mechanical properties such as diameter, length, shape, coating, and surface material (Jokstad et al. (2003), Esposito et al. (2014)).

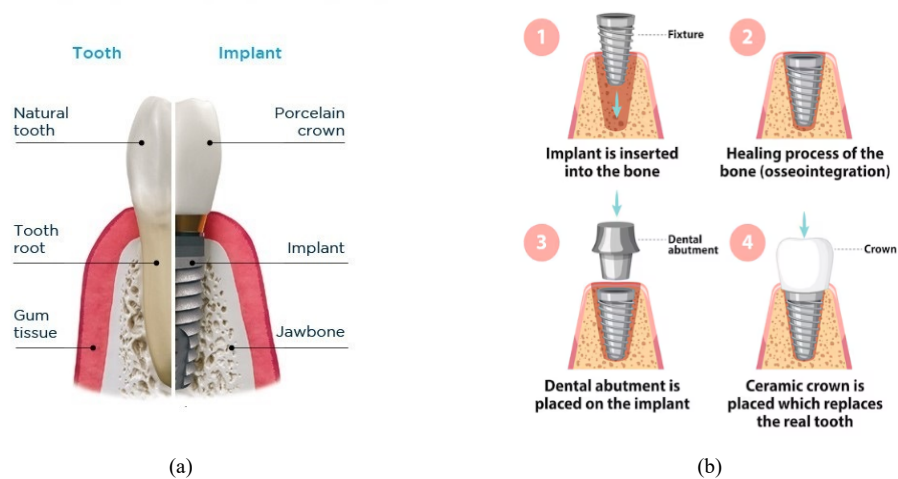


Figure 6.1 (a) Dental Implant Structure (LTD. (2024)) and (b) process of dental implant insertion (Burbank (2024))

Although dental implant applications are a rapidly increasing treatment option, many complications may occur after the treatment. Complications frequently encountered in implant-supported prostheses have been examined in many clinical

studies Chochlidakis et al. (2020), Moraschini et al. (2015), Papaspyridakos et al. (2020). Complications occur as mechanical, technical, and biological problems such as implant fixture or screw fractures, screw loosening, abutment fractures, or fractures in prosthetic superstructures (E. Jung et al. (2012); Lee et al. (2015); Lee et al. (2016)). It is important to detect these complications that occur over time in terms of both informing patients and guiding the treatment planning of dentists. In such cases, knowing the brand of the dental implant used is of paramount importance to address the issue and ensure continuity of treatment. Complications in implant supported prostheses are often related to abutments and attachments. Resolving these complications frequently requires the replacement of these components. In cases where a patient with a previously placed dental implant is unaware of the implant's brand and model, identifying the implant becomes a priority. Determining the implant brand by the dentist is a process that depends on experience and is time-consuming. If the implant brand cannot be accurately identified, attachments from similar implant manufacturers may be tested. However, this approach can damage the implant-attachment interface and lead to different complications. In addition, choosing and testing incorrect implant components causes extra costs. Individual parts can be produced by taking personal measurements for the existing implant, but this method also incurs additional time and costs. The final and avoided option is the removal of the implants from the jawbone without completing the prosthetic treatment. Considering these methods, accurately identifying the implant system enables the dentist to provide a high-quality solution to the patient, reduce the treatment duration, and avoid invasive procedures associated with implant removal.

In the absence of medical records, information about the type of implant can only be inferred from radiographs, as most implant fixtures are embedded in alveolar bone and cannot be observed during an oral examination. Therefore, radiographic identification of implants is particularly important for providing appropriate diagnosis and treatment to patients. However, identifying an implant brand based on imaging requires significant effort and expertise. Although this imaging is the primary tool for identifying dental implants, it is extremely difficult to distinguish between different

systems with similar implant types. In this thesis, considering the critical importance of identifying dental implant brands accurately, we proposed a novel Information Complexity-Receiver Operating Characteristic, abbreviated as ICOMP-ROC, to evaluate the performance of machine learning classification algorithms in identifying implant brands. Figure 6.2 illustrates the process steps of the flowchart for the study, detailing the analysis workflow on the dental image dataset.

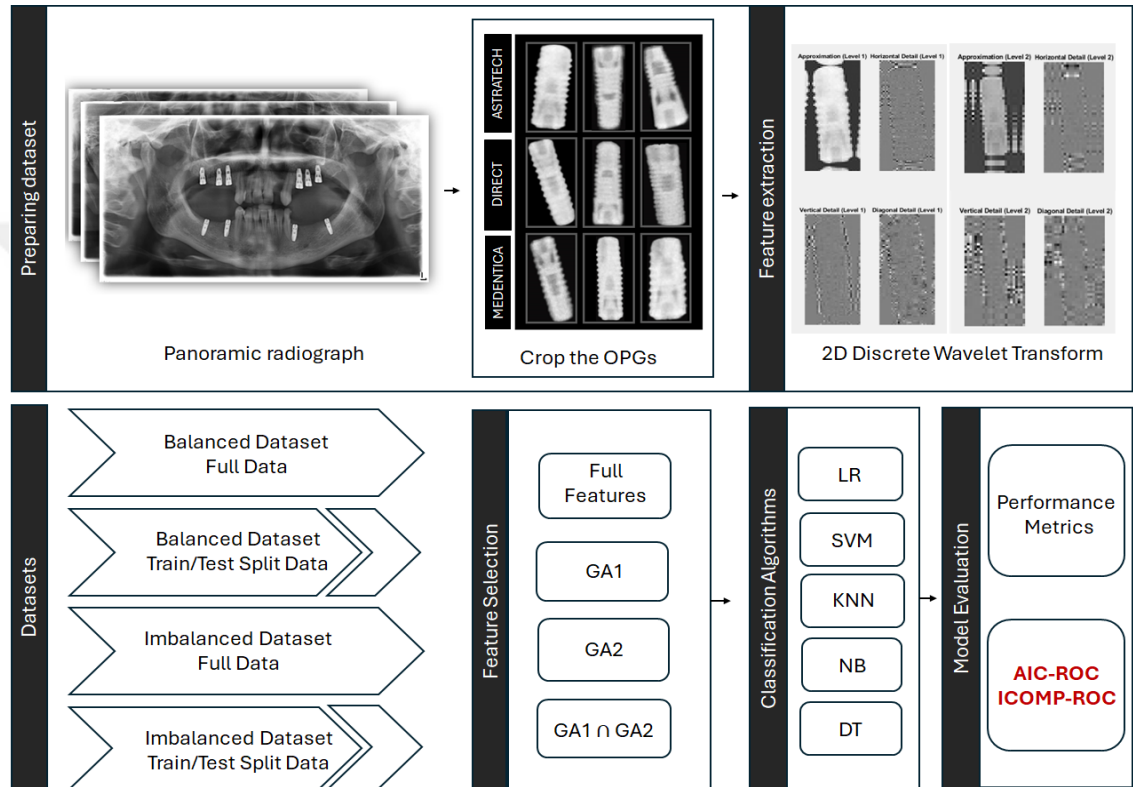


Figure 6.2 Flowchart of the process

6.1 Data Acquisition

In this study, orthopantomographs (OPG) of individuals who underwent dental implant treatment were obtained anonymously in JPEG format without including names or registration numbers from the imaging archiving systems of units within the Faculty of Dentistry at Gazi University between 2018 and 2022. Since this study follows a non-interventional retrospective design, no written or verbal informed consent was obtained from the participants. Ethical approval for the project was granted by the Non-Interventional Clinical Research Ethics Committee of Dokuz

Eylul University (Decision No. 2022/07-05, dated 23 Feb 2022). Images containing three types of implants were selected for this work: Astratech, Medentica, and Direct implants, focusing solely on the implant brands without considering variations in diameter, length, or width. The OPG data were labeled by the oral and maxillofacial surgeon based on electronic dental records and the dental implant usage register. During the labeling process, unique file names were created by coding the brand, file number, and implant positions as in Figure 6.3.

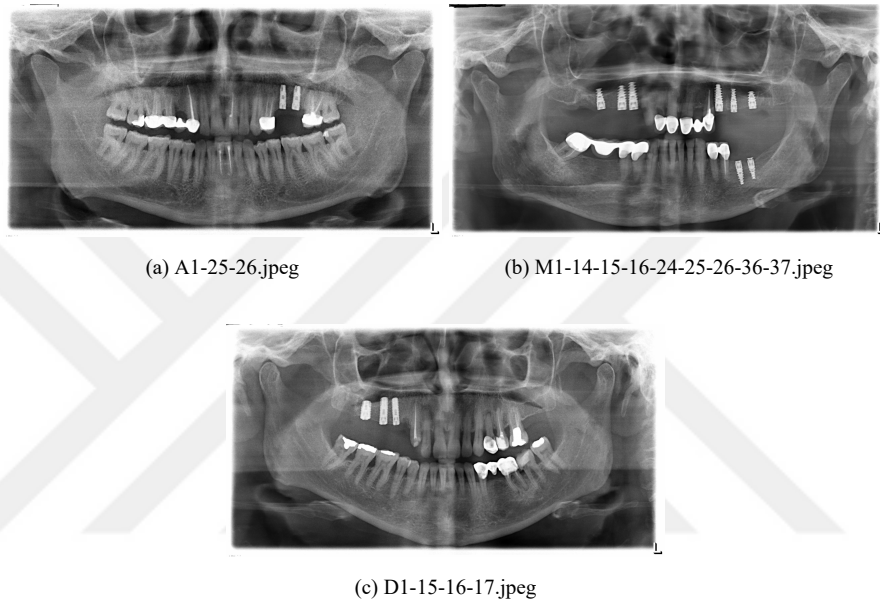


Figure 6.3 Anonymised OPG examples

Note: A1-25-26.jpeg: "A" represents Astratech, "1" refers to the file number, and "25, 26" denote the implant positions. M1-14-15-16-24-25-26-36-37.jpeg: "M" represents Medentica, "1" refers to the file number, and "14, 15, 16, 24, 25, 26, 36, 37" denote the implant positions. D1-15-16-17.jpeg: "D" represents Direct, "1" refers to the file number, and "15, 16, 17" denotes the implant position.

6.2 Pre-Processing

All obtained OPGs were subsequently reviewed by a prosthodontist. Images with severe noise, blurriness, or distortion that hindered the clinical identification and classification of implants were excluded. For the remaining OPGs, the relevant regions of implants were manually cropped using Adobe Photoshop, resulting in a dataset comprising Astratech, Medentica, and Direct implant images, carefully prepared for feature extraction. The types of dental implant systems and corresponding number of images are shown in Table 6.1.

Table 6.1 Number of OPGs and implants of implant types

	Number of OPGs	Number of Implants
Astratech	33	101
Medentica	142	108
Direct	37	141

6.3 Feature Extraction

We applied wavelet analysis proposed by Luis-Delgado et al. (2003) to our dataset. This method relies on the assumption that each implant group differs in terms of frequency (spectral) content. Therefore, the aim is to obtain the energy level of a sample and check if it is inside a band of frequency or not. The algorithm used for multiresolution decomposition is the Discrete Wavelet Transform (DWT) Mallat (1999). The DWT is equivalent to a series of low- and high-pass filtering of the original signal followed by down sampling. In case of a 2-D signal (i.e., implant image), high-pass filtering includes filtering in the horizontal, vertical, and diagonal directions. Thus, four images are generated at each level of decomposition: one low-pass and three high-pass (horizontal, vertical, and diagonal) images. In our study, three levels of DWT decomposition are performed for each sample image using biorthogonal wavelets. Mean, median, and variance of each level of decomposition are computed to obtain a 1×36 feature vector for each sample (Selver (2014)). To refine the dataset, 9 columns with values consistently equal to zero were removed from the features. This step eliminated redundant or non-informative features, ensuring that only meaningful features were retained, ultimately reducing the feature set to 1×27 feature vector for each sample.

6.4 Datasets and Feature Selection with GA

This section involves the creation of various datasets with different characteristics derived from the implant data to evaluate performance metrics and suggested information criteria for binary response modeling. Initially, a balanced dataset was created due to the similar sample sizes of Astratech (n=101) and Medentica (n=108)

implants. Additionally, an imbalanced dataset was constructed using Medentica (n=108) and others (n=242), which include Astratech (n=101) and Direct (n=141). Secondly, each balanced and imbalanced dataset was analyzed both as full data and as test/train data. The four cases are detailed below:

- Case 1: The balanced dataset with full data
- Case 2: The balanced dataset was split randomly into training (80%) and testing (20%) subsets
- Case 3: The imbalanced dataset with full data
- Case 4: The imbalanced dataset was split randomly into training (80%) and testing (20%) subsets

By implementing these cases, we aimed to analyze the models' performance comprehensively, comparing their behavior in situations where data is either fully utilized or randomly divided into training and testing sets.

Following these scenarios, Genetic Algorithms (GA) were employed to further evaluate the performance metrics in depth. To improve the performance of ML classification algorithms, two different GA methods, GA1 and GA2, were applied to both the full dataset and the train-test split dataset to identify the best subset of features, as given in Table Table 6.2. The main difference between the GA1 and GA2 approaches lies in GA2's ability to dynamically modify the crossover scheme in each generation, thereby enhancing computational efficiency. Additionally, the best subset of overlapping features obtained from both GA methods (i.e., $GA1 \cap GA2$) was selected to ensure consistency in the evaluation process and to enable a direct comparison of the classification algorithms under identical conditions.

Table 6.2 GA parameters

Parameter	GA1	GA2
Number of generations	50	50
Population size	100	100
Crossover probability	0.8	0.6
Mutation probability	0.3	0.001

6.5 Machine Learning Classification Algorithms

In this section, we employ fundamental classification algorithms including logistic regression (LR), support vector machines (SVM), Naive Bayes (NB), K-nearest neighbors (KNN) and decision trees (DT). A brief overview of these algorithms is as follows.

The LR model is a statistical modeling technique used to predict the probability of a dichotomous variable (e.g., 0 for the event not to occur or 1 for the event to happen) based on one or more input variables. It works by modeling the relationship between variables using a sigmoid function and a decision threshold for classification.

SVM is a supervised learning algorithm that finds an optimal hyperplane which maximally discriminates between various classes in an N-dimensional space. The hyperplane is described as a decision boundary that separates the two classes. The data points closest to the decision boundary, called support vectors, greatest influence on determining the location of the hyperplane. After identifying the hyperplane, new data are classified based on which side of the hyperplane it falls on.

The NB classification algorithm is a simple yet powerful probabilistic classifier based on Bayes's theorem with the naive feature independence assumption.

The KNN algorithm is a nonparametric method used in classification and regression. The working principle of this method is the assignment of data that is new in a previously created sample set to the cluster that has the closest (K) distance.

The DT classification is used to categorize data into distinct classes. The method constructs a tree-like model of decisions where each node represents a feature, each branch denotes the outcome of a decision, and each leaf node assigns a class label. This technique is visually intuitive, making it easy to understand and interpret. For a detailed explanation of the algorithms, see James et al. (2013).

6.6 Performance Metrics and Information-Based Criteria for Classification

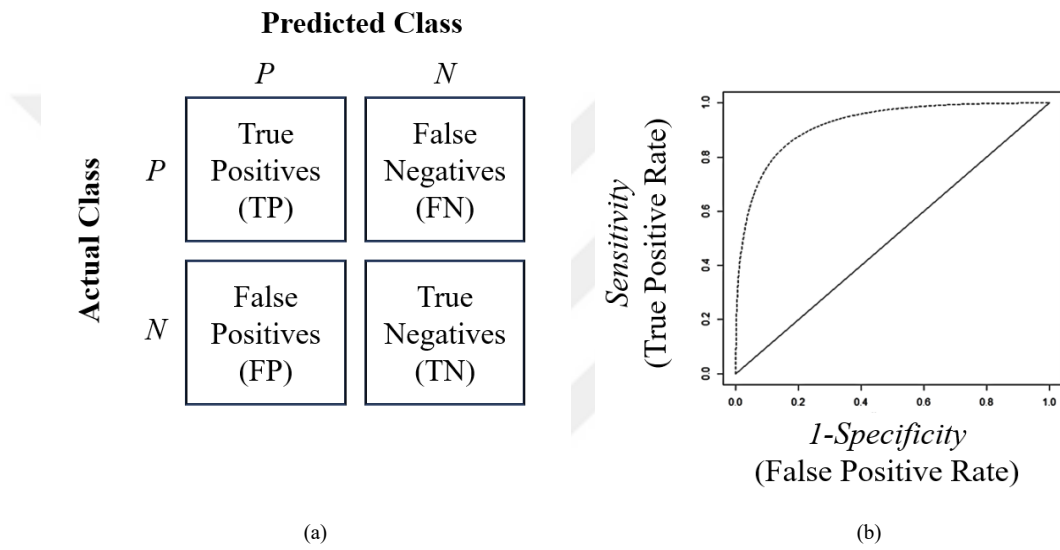


Figure 6.4 Illustration of Confusion Matrix (a) and ROC Curve (b)

A confusion matrix is a table that is often used to evaluate the performance of a classification model. It shows the number of correct and incorrect predictions made by the model compared with the actual outcomes in a test dataset. The rows represent the actual classes, and the columns represent the predicted test results in Figure 6.4a. Possibilities after cross-classification then include; a true positive (TP) occurs when the model correctly predicts the positive class, while a true negative (TN) when the model correctly predicts the negative class. Conversely, a false positive (FP) arises when the model incorrectly predicts the positive class, and a false negative (FN) occurs when the model incorrectly predicts the negative class. Using the confusion matrix, which tabulates the model's predictions against actual class labels, several performance metrics can be derived. These metrics include accuracy, precision, recall, specificity,

F1 score, and error rate. The metrics' equations are presented in Table 6.3.

Table 6.3 Performance Metrics

Performance metrics	Definition
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN} \quad (6.1)$
Precision	$\frac{TP}{TP+FP} \quad (6.2)$
Recall	$\frac{TP}{TP+FN} \quad (6.3)$
Specificity	$\frac{TN}{TN+FP} \quad (6.4)$
F1 Score	$\frac{2 \times (Recall \times Precision)}{Recall + Precision} \quad (6.5)$
Error Rate	$\frac{FP+FN}{TP+FN+FP+TN} \quad (6.6)$

Figure 6.4b shows the ROC curve, which is a graphical display of sensitivity (True Positive Rate) on the y-axis and 1—specificity (False Positive Rate) on the x-axis for varying cut-off points of test values. The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the ROC curve (AUC). The AUC ranges from 0 to 1, with 0.5 representing a random classifier and 1 representing a perfect classifier Hanley & McNeil (1982). In all performance metrics related to classification models (except for the error rate), higher values indicate better performance.

The computational form of ICOMP-ROC and AIC-ROC for the ML classification algorithm is the same as in Equation 3.15, and 3.19, except, here, in scoring ROC curves, we use the results from the ML classification algorithms to compute the probability estimates since actual response values and predicted response values are obtained. Then, these results are used in the standard ROC curve in the **perfcurve.m** function of MATLAB® to compute the correlation matrix R to score ICOMP-ROC for each of the ML classification algorithms.

6.7 Main Computational Results

All of our analysis were conducted using the MATLAB[®] computational platform, where the performance of classifiers was evaluated across four cases: balanced and imbalanced datasets, each assessed both with and without train-test splitting. Bold values in the results tables represent the best-performing classifier for each metric or model selection criterion.

6.7.1 Results of case 1: *Balanced Dataset with Full Data*

Table 6.4 presents the performance metrics and information-based model selection criteria for classification algorithms applied to the balanced dataset with full data with the full features, GA1, GA2, and $GA1 \cap GA2$ feature subsets.

According to the results presented in Table 6.4, in the Full Features and GA1 subsets, the best-performing model is Decision Tree (DT), demonstrating superior performance across all metrics and information-based criteria. In contrast, Naïve Bayes (NB) is the worst-performing model in these subsets. For the GA2 subset, DT remains the best model across all metrics and information-based criteria. However, in this subset, the worst-performing models vary depending on the specific metric. Naïve Bayes (NB) has the lowest accuracy (0.7368), recall (0.5926), specificity (0.6995), and highest error rate (0.2632), AIC-ROC (-111.4200), and ICOMP-ROC (-114.1000), while K-Nearest Neighbors (KNN) exhibits the weakest performance in terms of precision (0.8198) and F1-score (0.8020). According to $GA1 \cap GA2$, DT remains the best model for accuracy (0.9139), recall (0.9444), specificity (0.9189), and AUC (0.9582). The error rate is relatively low (0.0861), while AIC-ROC (-360.3000) and ICOMP-ROC (-364.0600) are the best values in this subset. However, within this group, SVM outperforms DT in precision (0.9146) and F1-score (0.9307), making it the best-performing model for these specific metrics. On the other hand, Naïve Bayes (NB) performs the worst overall, with the lowest accuracy (0.7464), recall (0.5926), and specificity (0.7072), along with the highest error rate (0.2536), AIC-ROC (-114.2300), and ICOMP-ROC (-117.0000),

indicating its weak classification performance. Additionally, KNN and DT exhibits the lowest F1-score (0.8812), making it another weak performer in this subset. Overall, in terms of accuracy, the best model is GA1 - DT (0.9426). For precision, the best model is Full Features - DT (0.9612). In recall, GA1 - DT and GA1 $\cap$ GA2 - DT perform best (0.9444), while the best specificity is observed in GA1 - DT (0.9444). Regarding F1-score, Full Features - DT achieves the highest performance (0.9604). Finally, in terms of information-based criteria, GA1 - DT achieves the best AIC-ROC (-428.3000) and ICOMP-ROC (-432.0800), making it the most optimal model overall.

Table 6.4 Comparison of performance metrics and model selection criteria for balanced dataset with full data

		Accuracy	Precision	Recall	Specificity	F1Score	Error Rate	AUC	AIC-ROC	ICOMP-ROC
Full Features	LR	0.8325	0.8411	0.8333	0.8372	0.8317	0.1675	0.8975	-217.7900	-220.8500
	SVM	0.8469	0.9043	0.7870	0.8416	0.9109	0.1531	0.9016	-189.2200	-192.4400
	NB	0.7560	0.7938	0.7130	0.7512	0.8020	0.2440	0.8030	-102.6500	-105.1300
	KNN	0.8421	0.8378	0.8611	0.8493	0.8218	0.1579	0.9057	-220.6900	-223.8000
	DT	0.9378	0.9612	0.9167	0.9384	0.9604	0.0622	0.9638	-400.2100	-403.9200
GA1	LR	0.8086	0.8696	0.7407	0.8000	0.8812	0.1914	0.8635	-179.8400	-182.8300
	SVM	0.8134	0.9157	0.7037	0.7958	0.9307	0.1866	0.8667	-169.4100	-172.5400
	NB	0.7081	0.8133	0.5648	0.6667	0.8614	0.2919	0.7759	-92.2180	-94.8550
	KNN	0.8565	0.8980	0.8148	0.8544	0.9010	0.1435	0.9336	-262.5200	-265.9300
	DT	0.9426	0.9444	0.9444	0.9444	0.9406	0.0574	0.9795	-428.3000	-432.0800
GA2	LR	0.8469	0.8800	0.8148	0.8462	0.8812	0.1531	0.8986	-216.5900	-219.7100
	SVM	0.8469	0.9130	0.7778	0.8400	0.9208	0.1531	0.9026	-200.9800	-204.1700
	NB	0.7368	0.8533	0.5926	0.6995	0.8911	0.2632	0.8032	-111.4200	-114.1000
	KNN	0.8230	0.8198	0.8426	0.8311	0.8020	0.1770	0.9148	-233.1600	-236.3800
	DT	0.9330	0.9434	0.9259	0.9346	0.9406	0.0670	0.9769	-417.9800	-421.8400
GA1 $\cap$ GA2	LR	0.8182	0.8977	0.7315	0.8061	0.9109	0.1818	0.8721	-177.9200	-180.9700
	SVM	0.8086	0.9146	0.6944	0.7895	0.9307	0.1914	0.8622	-167.1300	-170.2200
	NB	0.7464	0.8767	0.5926	0.7072	0.9109	0.2536	0.7818	-114.2300	-117.0000
	KNN	0.8565	0.8824	0.8333	0.8571	0.8812	0.1435	0.9351	-264.8700	-268.3100
	DT	0.9139	0.8947	0.9444	0.9189	0.8812	0.0861	0.9582	-360.3000	-364.0600

Note: Full Features{1:27}, GA1{1, 5, 8, 9, 12, 14, 16, 17, 20, 23, 25, 26, 27}, GA2{1, 3, 4, 6, 9, 10, 12, 13, 15, 16, 17, 18, 20, 22, 25, 26, 27}, GA1 $\cap$ GA2{1, 9, 12, 17, 20, 25, 26, 27}

Figure 6.5 compares the ROC curves for various classification methods across four feature subsets: (a) Full Features, (b) GA1, (c) GA2, and (d) GA1 $\cap$ GA2, for the balanced dataset with full data. The AUC values for each classifier are indicated in the

legends. Across all subsets, DT consistently achieves the highest AUC values, while NB performs the worst with the lowest AUC values.

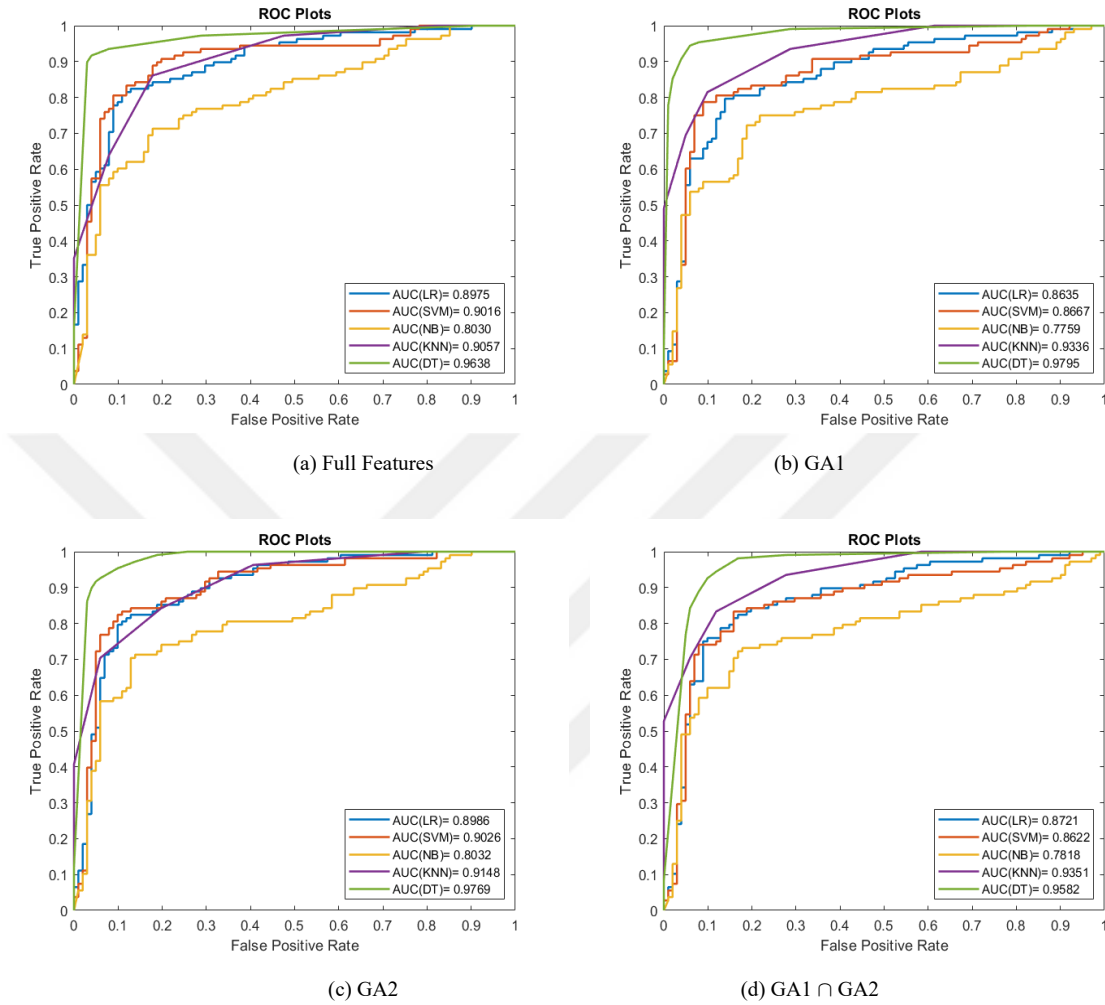


Figure 6.5 Comparison of classification methods ROC curves for balanced dataset with full data

6.7.2 Results of case 2: Balanced Dataset with Train/Test Split Data

Table 6.5 presents the performance metrics and information-based model selection criteria for classification algorithms applied to the balanced dataset with train-test splitting.

According to Table 6.5, for the Full Features, SVM performs the best in terms of accuracy (0.7143), precision (0.7368), F1 score (0.7000), error rate (0.2857), AUC (0.6712), AIC-ROC (-423.3700), and ICOMP-ROC (-426.0900). LR is the best for

Table 6.5 Comparison of performance metrics and model selection criteria for balanced dataset with train/test split data

		Accuracy	Precision	Recall	Specificity	F1Score	Error Rate	AUC	AIC-ROC	ICOMP-ROC
	LR	0.6905	0.6818	0.7143	0.6667	0.6977	0.3095	0.6621	-400.4900	-403.1300
	SVM	0.7143	0.7368	0.6667	0.7619	0.7000	0.2857	0.6712	-423.3700	-426.0900
	NB	0.6429	0.6875	0.5238	0.7619	0.5946	0.3571	0.6485	-368.2600	-370.4100
	KNN	0.4762	0.4800	0.5714	0.3810	0.5217	0.5238	0.5261	-366.8300	-367.0000
	DT	0.6191	0.6316	0.5714	0.6667	0.6000	0.3810	0.6689	-358.6800	-360.5700
GA1	LR	0.6667	0.5625	0.5625	0.7308	0.5625	0.3333	0.7428	-461.8000	-463.9300
	SVM	0.7857	0.8182	0.5625	0.9231	0.6667	0.2143	0.7909	-491.9800	-494.3300
	NB	0.5238	0.4286	0.7500	0.3846	0.5455	0.4762	0.7188	-369.5200	-372.0700
	KNN	0.7143	0.6000	0.7500	0.6923	0.6667	0.2857	0.7560	-452.8600	-455.0700
	DT	0.5952	0.4800	0.7500	0.5000	0.5854	0.4048	0.6827	-345.5200	-347.9400
GA2	LR	0.7619	0.8000	0.7273	0.8000	0.7619	0.2381	0.8500	-503.1000	-505.9300
	SVM	0.8810	0.9474	0.8182	0.9500	0.8781	0.1191	0.9114	-577.8400	-581.0000
	NB	0.7619	0.7308	0.8636	0.6500	0.7917	0.2381	0.8932	-498.4400	-501.6500
	KNN	0.9048	0.9091	0.9091	0.9000	0.9091	0.0952	0.9205	-592.0000	-595.5200
	DT	0.8571	0.8636	0.8636	0.8500	0.8636	0.1429	0.8602	-539.9500	-543.0800
GA1 $\cap$ GA2	LR	0.8810	0.8824	0.8333	0.9167	0.8571	0.1191	0.8657	-530.7200	-534.0500
	SVM	0.8095	0.8571	0.6667	0.9167	0.7500	0.1905	0.8519	-519.7500	-523.0400
	NB	0.7143	0.8000	0.4444	0.9167	0.5714	0.2857	0.7500	-417.8200	-420.5800
	KNN	0.7857	0.8000	0.6667	0.8750	0.7273	0.2143	0.8241	-511.7000	-514.4800
	DT	0.8095	0.8125	0.7222	0.8750	0.7647	0.1905	0.7986	-483.2900	-486.6600

Note: Full Features {1:27}, GA1 {1, 4, 6, 7, 8, 9, 17, 19, 21, 23, 25, 26, 27}, GA2 {2, 6, 8, 11, 13, 15, 17, 19, 20, 21, 22, 23, 25, 27}, GA1 $\cap$ GA2 {6, 17, 19, 21, 23, 25, 27}

recall (0.7143), while both SVM and NB perform equally well for specificity (0.7619). For the GA1, SVM is the best algorithm for accuracy (0.7857), precision (0.8182), specificity (0.9231), error rate (0.2143), AUC (0.7909), AIC-ROC (-491.9800), and ICOMP-ROC (-494.3300). Recall is best achieved by NB, KNN, and DT equally (0.7500), while SVM and KNN both excel in F1 score (0.6667). In the GA2, KNN demonstrates the best performance for accuracy (0.9048), recall (0.9091), F1 score (0.9091), error rate (0.0952), AUC (0.9205), AIC-ROC (-592.0000), and ICOMP-ROC (-595.5200), while SVM is the best for precision (0.9474) and specificity (0.9500). Lastly, in the GA1 $\cap$ GA2 subset, LR outperforms others in accuracy (0.8810), precision (0.8824), recall (0.8333), specificity (0.9167), F1 score (0.8571), error rate (0.1191), AUC (0.8657), AIC-ROC (-530.7200), and ICOMP-ROC (-534.0500). For the balanced dataset with train-test splitting, the KNN classifier with the GA2 subset delivers the best overall performance, according to

information-based criteria and all performance metrics except precision and specificity.

Figure 6.6 compares the ROC curves for various classification methods across four feature subsets: (a) Full Features, (b) GA1, (c) GA2, and (d) $GA1 \cap GA2$, for the dataset with train-test split data. The AUC values for each classifier are indicated in the legends. The highest AUC value is obtained from GA2 with KNN (AUC = 0.9205).

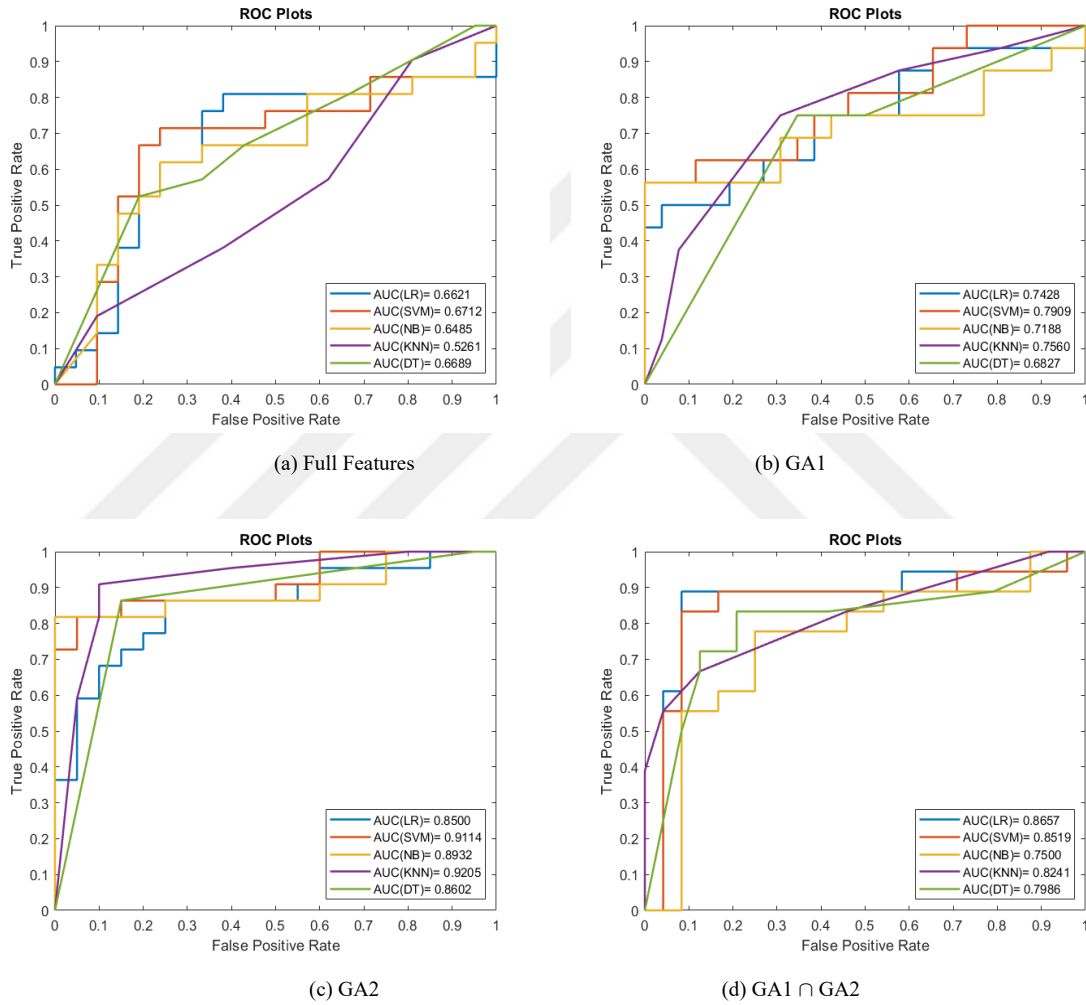


Figure 6.6 Comparison of classification methods ROC curves for dataset with train/test split data

6.7.3 Results of case 3: Imbalanced Dataset with Full Data

Table 6.6 presents the performance metrics and information-based model selection criteria for various classification algorithms applied to the imbalanced dataset with the

full feature set, GA1, GA2, and $GA1 \cap GA2$ feature subsets.

Table 6.6 Comparison of performance metrics and model selection criteria for imbalanced dataset with full data

		Accuracy	Precision	Recall	Specificity	F1Score	Error Rate	AUC	AIC-ROC	ICOMP-ROC
Full Features	LR	0.8686	0.8523	0.6944	0.7653	0.9463	0.1314	0.9055	-445.9200	-449.0600
	SVM	0.8800	0.9024	0.6852	0.7790	0.9669	0.1200	0.8852	-399.8000	-402.9200
	NB	0.7829	0.6455	0.6574	0.6514	0.8388	0.2171	0.8012	-221.4700	-223.7200
	KNN	0.8457	0.7935	0.6759	0.7300	0.9215	0.1543	0.9154	-429.0700	-432.2300
	DT	0.9600	0.9608	0.9074	0.9333	0.9835	0.0400	0.9770	-845.0700	-848.8800
GA1	LR	0.8429	0.8732	0.5741	0.6927	0.9628	0.1571	0.8654	-377.3200	-380.3000
	SVM	0.8514	0.8889	0.5926	0.7111	0.9669	0.1486	0.8645	-360.8300	-363.9000
	NB	0.7771	0.6250	0.6944	0.6579	0.8141	0.2229	0.7992	-238.6700	-241.0900
	KNN	0.8829	0.8526	0.7500	0.7980	0.9422	0.1171	0.9436	-504.0000	-507.4300
	DT	0.9629	0.9612	0.9167	0.9384	0.9835	0.0371	0.9817	-863.2000	-867.0500
GA2	LR	0.8743	0.8902	0.6759	0.7684	0.9628	0.1257	0.8961	-429.0100	-432.1400
	SVM	0.8543	0.9014	0.5926	0.7151	0.9711	0.1457	0.8727	-375.5400	-378.5500
	NB	0.7857	0.6602	0.6296	0.6446	0.8554	0.2143	0.7921	-243.0300	-245.3400
	KNN	0.8857	0.8617	0.7500	0.8020	0.9463	0.1143	0.9358	-486.7900	-490.1800
	DT	0.9514	0.9418	0.8982	0.9194	0.9752	0.0486	0.9873	-812.2600	-816.1400
$GA1 \cap GA2$	LR	0.8400	0.8714	0.5648	0.6854	0.9628	0.1600	0.8655	-376.6600	-379.6300
	SVM	0.8343	0.8677	0.5463	0.6705	0.9628	0.1657	0.8657	-361.4400	-364.4900
	NB	0.7686	0.6116	0.6852	0.6463	0.8058	0.2314	0.7961	-262.0300	-264.6300
	KNN	0.8686	0.8370	0.7130	0.7700	0.9380	0.1314	0.9443	-504.4600	-507.9000
	DT	0.9543	0.9510	0.8982	0.9238	0.9793	0.0457	0.9781	-770.3300	-774.1300

Note: Full Features{1:27}, GA1{1, 2, 4, 5, 6, 12, 14, 17, 19, 21, 22, 27}, GA2{1, 2, 3, 5, 9, 12, 19, 22, 24, 25, 26, 27}, $GA1 \cap GA2$ {1, 2, 5, 12, 19, 22, 27}

According to Table 6.6, the DT is the best-performing classification algorithm across full features, GA1, GA2, and $GA1 \cap GA2$, based on both performance metrics and information-based criteria. In contrast, NB performs the worst across these subsets. For the GA1, DT achieves the highest accuracy (0.9629), precision (0.9612), recall (0.9167), specificity (0.9384), and the lowest error rate (0.0371), AIC-ROC (-863.2000), and ICOMP-ROC (-867.0500). For F1 scores, DT achieves the best value for both the full features (0.9835) and GA1 (0.9835). Additionally, DT delivers a high AUC (0.9873) in the GA2 subset. In conclusion, for the imbalanced dataset with full data, the DT combined with the GA1 subset emerges as the most effective solution for classification, considering all performance metrics (except AUC) and

information-based criteria.

Figure 6.7 compares the ROC curves for various classification methods across four feature subsets: (a) Full Features, (b) GA1, (c) GA2, and (d) $GA1 \cap GA2$, for the imbalanced dataset with full data. The AUC values for each classifier are indicated in the legends. Across all subsets, DT has the highest AUC value in the GA2 (AUC = 0.9873). NB performs the worst with the lowest AUC values across all subsets for imbalanced data.

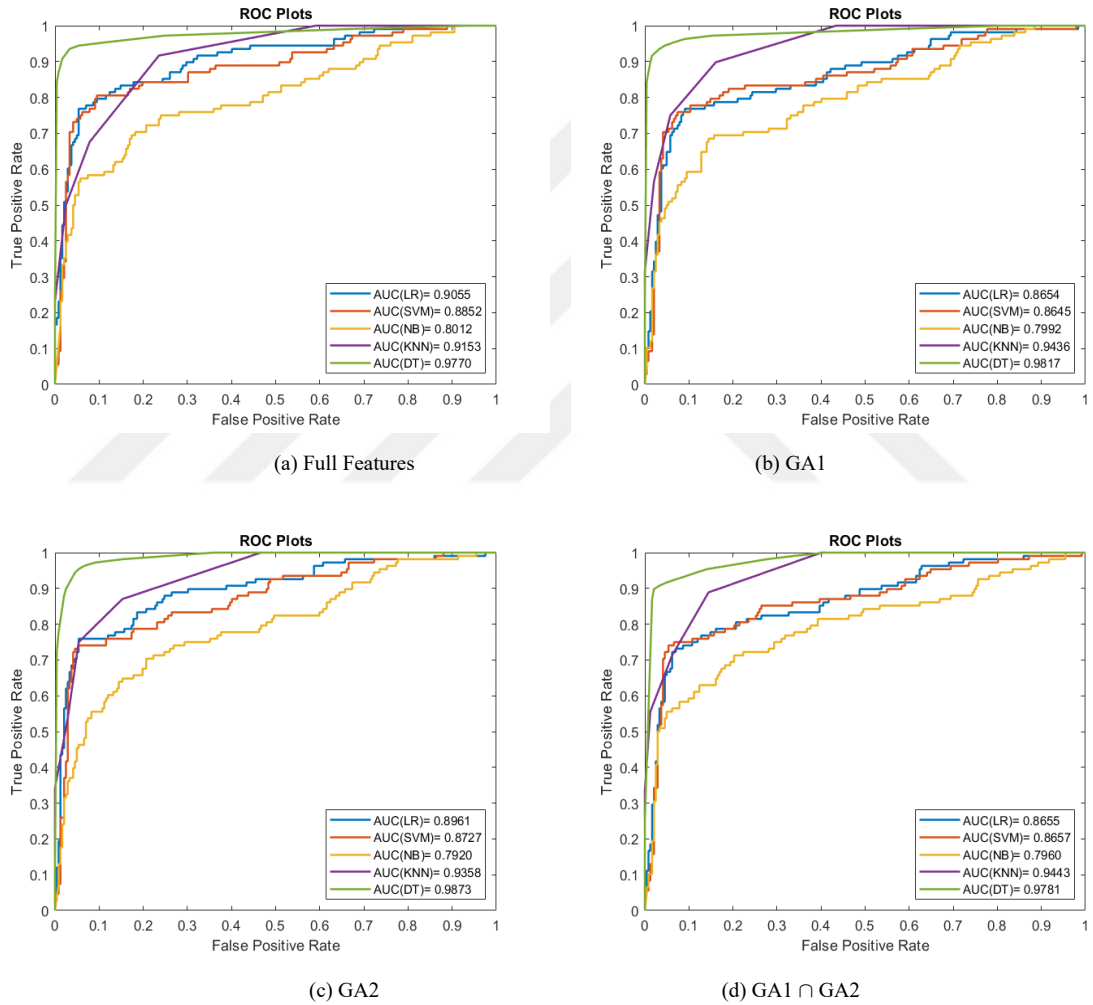


Figure 6.7 Comparison of classification methods ROC curves for imbalanced dataset with full data

6.7.4 Results of case 4: Imbalanced Dataset with Train/Test Split Data

Table 6.7 presents the performance metrics and information-based model selection criteria for classification algorithms applied to the imbalanced dataset with train-test splitting.

Table 6.7 Comparison of performance metrics and model selection criteria for imbalanced dataset with train/test split data

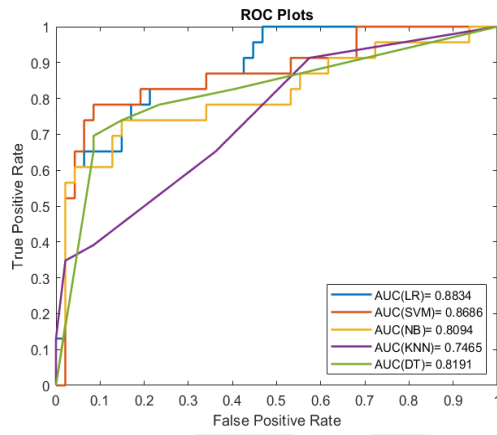
		Accuracy	Precision	Recall	Specificity	F1Score	Error Rate	AUC	AIC-ROC	ICOMP-ROC
Full Features	LR	0.8286	0.7895	0.6522	0.9149	0.7143	0.1714	0.8834	-937.1400	-940.0600
	SVM	0.8571	0.8824	0.6522	0.9575	0.7500	0.1429	0.8686	-953.5100	-956.5700
	NB	0.8000	0.7368	0.6087	0.8936	0.6667	0.2000	0.8094	-830.2100	-832.8500
	KNN	0.7429	0.6923	0.3913	0.9149	0.5000	0.2571	0.7465	-810.6000	-812.7300
	DT	0.8429	0.8000	0.6957	0.9149	0.7442	0.1571	0.8192	-871.4800	-874.9100
GA1	LR	0.8714	0.8261	0.7917	0.9130	0.8085	0.1286	0.8841	-914.6700	-917.8500
	SVM	0.8286	0.8333	0.6250	0.9348	0.7143	0.1714	0.8922	-912.1100	-915.3600
	NB	0.7857	0.6957	0.6667	0.8478	0.6809	0.2143	0.8546	-819.9200	-822.9300
	KNN	0.7286	0.6000	0.6250	0.7826	0.6122	0.2714	0.7604	-799.6200	-801.9400
	DT	0.7571	0.6296	0.7083	0.7826	0.6667	0.2429	0.8034	-812.2100	-815.5400
GA2	LR	0.8714	0.9333	0.6364	0.9792	0.7568	0.1286	0.8352	-943.5400	-946.5500
	SVM	0.8857	0.9375	0.6818	0.9792	0.7895	0.1143	0.8343	-961.7600	-964.8700
	NB	0.8143	0.7647	0.5909	0.9167	0.6667	0.1857	0.8438	-884.5000	-887.3100
	KNN	0.8714	0.8824	0.6818	0.9583	0.7692	0.1286	0.8542	-938.3100	-941.4000
	DT	0.7571	0.5926	0.7273	0.7708	0.6531	0.2429	0.7311	-754.0900	-757.3900
GA1 $\cap$ GA2	LR	0.8286	1.0000	0.5200	1.0000	0.6842	0.1714	0.8613	-915.1600	-918.0600
	SVM	0.8286	1.0000	0.5200	1.0000	0.6842	0.1714	0.8738	-918.3000	-921.3200
	NB	0.8000	1.0000	0.4400	1.0000	0.6111	0.2000	0.8222	-841.3400	-843.9600
	KNN	0.8429	0.8500	0.6800	0.9333	0.7556	0.1571	0.8529	-946.4300	-949.7000
	DT	0.7714	0.6957	0.6400	0.8444	0.6667	0.2286	0.7849	-802.6400	-805.8700

Note: Full Features{1:27}, GA1{2, 3, 4, 10, 14, 15, 19, 21, 22, 26, 27}, GA2{3, 9, 11, 16, 22, 23, 24, 26, 27}, GA1 $\cap$ GA2{3, 22, 26, 27}

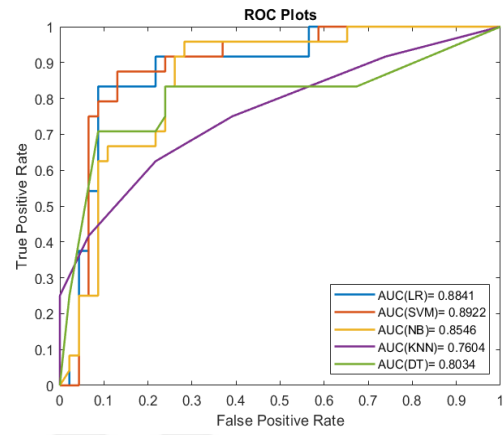
Based on Table 6.7, for the full features, the best algorithm is SVM for accuracy (0.8571), precision (0.8824), specificity (0.9575), F1 score (0.7500), AIC-ROC (-953.5100), and ICOMP-ROC (-956.5700). For recall (0.6957) and error rate (0.1571), DT performs the best, while LR leads in AUC (0.8834). In the GA1, LR achieves the best performance for accuracy (0.8714), recall (0.7917), F1 score (0.8085), error rate (0.1286), AIC-ROC (-914.6700), and ICOMP-ROC (-917.8500).

However, SVM performs best for precision (0.8333), specificity (0.9348), and AUC (0.8922). For the GA2, SVM performs the best for accuracy (0.8857), precision (0.9375), F1 score (0.7895), error rate (0.1143), AIC-ROC (-961.7600), and ICOMP-ROC (-964.8700). The DT algorithm demonstrates the highest level of recall (0.7273), while LR and SVM exhibit equivalent specificity levels of 0.9792. In the $GA1 \cap GA2$, KNN achieves the highest accuracy (0.8429) and recall (0.6800). Precision (1.0000) and specificity (1.0000) are shared by LR, SVM, and NB. SVM performs best for AUC (0.8738), while KNN is the best for F1 score (0.7556), AIC-ROC (-946.4300) and ICOMP-ROC (-949.7000). When analyzed based on accuracy, GA2 with SVM shows the best performance. For specificity and precision, the $GA1 \cap GA2$ highlights LR, SVM, and NB as the top algorithms. In terms of recall and F1 score, GA1 with LR performs the best. For error rate, AIC-ROC, and ICOMP-ROC, GA2 with SVM demonstrates best results, while AUC is highest for GA1 with SVM.

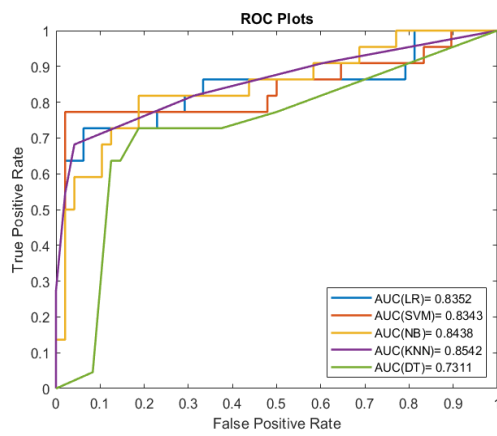
Figure 6.8 compares the ROC curves for various classification methods across four feature subsets: (a) Full Features, (b) GA1, (c) GA2, and (d) $GA1 \cap GA2$, for the imbalanced dataset with train-test split data. The AUC values for each classifier are indicated in the legends. Across all subsets, SVM achieves the highest AUC value in the GA1 subset (AUC = 0.8922).



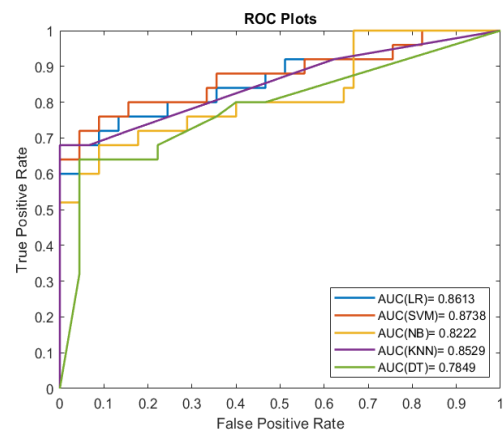
(a) Full Features



(b) GA1



(c) GA2



(d) GA1 ∩ GA2

Figure 6.8 Comparison of classification methods ROC curves for imbalanced dataset with train/test split data

CHAPTER SEVEN

CONCLUSIONS AND DISCUSSION

In this thesis, new ICOMP-ROC and AIC-ROC information criteria were proposed for model selection to choose the best Bi-distributional ROC curve model among a portfolio of Bi-distributions. More specifically, sixteen different Bi-distribution combinations were constructed in a large-scale Monte Carlo simulation, and the performance of ICOMP-ROC and AIC-ROC criteria was empirically compared with the widely known AUC (Area Under the Curve) from the literature to choose the best fitting Bi-distribution. Simulation results were based on an imbalanced dataset. That is, different sample sizes were used for the negative class (X_1) and positive class (X_2). From Case 1 of the simulation experiment, the Weibull-Gamma distribution was mixed as a pair of pseudo-true distribution. In this scenario, results showed that ICOMP-ROC and AIC-ROC criteria correctly identified the pseudo-true Weibull-Gamma distribution pair; in contrast, the AUC incorrectly identified the Weibull-Exponential pair. In addition, in Case 2 of the simulation experiment, results showed that the pseudo-true Normal-Normal distribution pair achieved a 100% success hit rate using ICOMP-ROC compared to AUC, which identified the GP-Normal distribution pair, and AIC-ROC identified the Exponential-Exponential distribution pair as the best fit, with a 100% success rate, but failed to recognize the correct distribution as the Normal-Normal distribution pair. This is not surprising for the performance of the AUC, even though it is a popular metric in ROC curve modeling. As discussed in Halligan et al. (2015), AUC has some drawbacks and limitations. The computational cost of AUC is high since AUC is inherently dependent upon the estimation of the ROC curve. The computational complexity of AUC is high for multiclass problems, as reported in Hand & Till (2001), Provost & Domingos (2003). Although simulation studies were relied on in choosing the best Bi-distributions in ROC curve modeling in the first part of the thesis, the proposed approach can be easily applied to real dataset scenarios where Bi-distributional fitting is required for the negative class (X_1) and positive class (X_2).

Encouraged by the results of the Bi-distributional fitting, in the second part of the thesis, several well-known classification machine learning (ML) algorithms were evaluated using orthopantomography (OPG) data from individuals undergoing dental implant treatment. More specifically, logistic regression (LR), support vector machines (SVM), naive Bayes (NB), k-nearest neighbors (KNN), and decision trees (DT) were evaluated. Datasets were created so that Astaratech and Medentica data are balanced, while Medentica and others represent unbalanced data. This approach allowed for a comprehensive evaluation of the impact of data balance on the outcomes for each case scenario, ensuring both scenarios were thoroughly investigated. The purpose in this part of the thesis was to determine how to select an optimal ML classification model. As is well-known, there are many traditional performance evaluation measures when it comes to selecting a classification model. The performance of newly proposed novel information-based criteria, namely, ICOMP-ROC and AIC-ROC, was studied along with other traditional classification performance metrics. From a practical point of view, the goal was to choose a classification algorithm with the best predictive performance on real datasets considered.

In all the ML classification algorithms, the information regarding the performance of these algorithms is summarized into a confusion matrix. This matrix is built by comparing the observed and predicted classes for a set of observations. It contains all the information needed to calculate most of the traditional classification performance metrics, such as Accuracy, Precision, Recall, F1 Score, Error Rate, and others.

While attempting to determine the true distribution pair for TPR and FPR, and to identify the best ML classification algorithm for real data, it was observed that performance metrics introduced challenges in the decision-making process. The inconsistencies and differing recommendations provided by each performance metric made it difficult to arrive at a clear conclusion regarding the optimal choice. However, it was shown that ICOMP-ROC provides a clearer and more reliable assessment of classifier performance by effectively integrating both predictive accuracy and model complexity, thus supporting a more informed decision-making

process in classifier selection. As noted in the literature, the basic key difference between the ICOMP criterion and AIC is ICOMP's inclusion of complexity, which takes into account the correlation structure in the parameter estimates. Although in the classification results, ICOMP-ROC and AIC-ROC are minimized at the same ML classifier, the fact that ICOMP-ROC, due to its ability to account for model complexity through the complexity of the celebrated inverse Fisher information matrix, provides a more robust criterion in high-dimensions and eliminates counting and penalizing the number of parameters in the model explicitly was emphasized.

Additionally, the use of Genetic Algorithms (GA) in this study plays a crucial role in optimizing the feature selection process. GA was employed to reduce the dimensionality of the datasets by identifying the most relevant features that contribute to classification accuracy. By mimicking the process of natural selection, GA iteratively refines the feature set, improving classification performance while reducing computational complexity. The usage of GA is particularly important in this study as it helps minimize the model's complexity, which is essential for more efficient and interpretable models, thereby helping to improve the accuracy of ML classification algorithms. It is recognized that in some complex data problems, GA can be slow and may require speeding up. Further, it is acknowledged that ROC curve modeling is not suitable for all datasets. The study of the impact of class imbalance on classification performance has been undertaken by Luque et al. (2019). The approach can also handle imbalanced data as illustrated in the Monte Carlo simulation study as well as in the real datasets.

There are other bivariate probability distributions to consider to score the ROC curve other than the bivariate normal distribution to guard against the non-Gaussianity in datasets. In future studies, other bivariate probability distributions will be considered along with relaxing the Bi-distributional assumption, and the performance of the nonparametric approach to ROC curve modeling using the kernel density estimation (KDE) approach will be studied. Further, binary classification results will be generalized to multi-class classification problems when more than two groups are present.

REFERENCES

- Akaike, H. (1973). Information theory and an extension of maximum likelihood principle. In *Proc. 2nd Int. Symp. on Information Theory*, (267–281).
- Bayes, T. (1763). Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53), 370–418.
- Bozdogan, H. (1987). Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3), 345–370.
- Bozdogan, H. (1988). Icomp: A new model selection criterion, in classification and related methods of data analysis. *HH Bock (ed.)*.
- Bozdogan, H. (1990a). Multisample cluster analysis of the common principal component model in k groups using an entropic statistical complexity criterion, invited paper presented at the international symposium on theory and practice of classification. *Puschino, Soviet Union*.
- Bozdogan, H. (1990b). On the information-based measure of covariance complexity and its application to the evaluation of multivariate linear models. *Communications in Statistics-Theory and Methods*, 19(1), 221–278.
- Bozdogan, H. (1993). Choosing the number of component clusters in the mixture-model using a new informational complexity criterion of the inverse-fisher information matrix. In *Information and classification* (40–54). Springer.
- Bozdogan, H. (1994). Choosing the number of clusters, subset selection of variables, and outlier detection in the standard mixture-model cluster analysis. In *New approaches in classification and data analysis* (169–177). Springer.
- Bozdogan, H. (2000). Akaike's information criterion and recent developments in information complexity. *Journal of mathematical psychology*, 44(1), 62–91.
- Bozdogan, H. (2003). Intelligent statistical data mining with information complexity and genetic algorithms hamparsum bozdogan university of tennessee, knoxville,

- usa. In *Statistical data mining and knowledge discovery* (47–88). Chapman and Hall/CRC.
- Breiman, L. (2017). *Classification and regression trees*. Routledge.
- Burbank, M. D. (2024). Implant crown types burbank. Accessed: 24-Dec-2024.
- Calvetti, D., Golub, G., Gragg, W., & Reichel, L. (2000). Computation of gauss-kronrod quadrature rules. *Mathematics of computation*, 69(231), 1035–1052.
- Chochlidakis, K., Einarsdottir, E., Tsigarida, A., Papaspyridakos, P., Romeo, D., Barmak, A. B., & Ercoli, C. (2020). Survival rates and prosthetic complications of implant fixed complete dental prostheses: an up to 5-year retrospective study. *The Journal of prosthetic dentistry*, 124(5), 539–546.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20, 273–297.
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21–27.
- Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2), 215–232.
- E. Jung, R., Zembic, A., Pjetursson, B. E., Zwahlen, M., & S. Thoma, D. (2012). Systematic review of the survival rate and the incidence of biological, technical, and aesthetic complications of single crowns on implants reported in longitudinal studies with a mean follow-up of 5 years. *Clinical oral implants research*, 23, 2–21.
- Egan, J. P. (1975). Signal detection theory and roc analysis. (*No Title*).
- Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Bristol:Springer.
- Esposito, M., Ardebili, Y., & Worthington, H. V. (2014). Interventions for replacing missing teeth: different types of dental implants. *Cochrane database of systematic reviews*, (7).

- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2), 179–188.
- Fix, E., & Hodges Jr, J. (1951). Discriminatory analysis-nonparametric discrimination: consistency properties. technical report project 21-49-004. Technical report, Report.
- Gauss, C. F. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. sumtibus Frid. Perthes et IH Besser.
- Gneiting, T., & Vogel, P. (2022). Receiver operating characteristic (roc) curves: equivalences, beta model, and minimum distance estimation. *Machine learning*, 111(6), 2147–2159.
- Gneiting, T., & Walz, E.-M. (2022). Receiver operating characteristic (roc) movies, universal roc (uroc) curves, and coefficient of predictive ability (cpa). *Machine Learning*, 111(8), 2769–2797.
- Goldberg, D. E. (1989). Genetic algorithm in search, optimization and machine learning, addison. *Wesley Publishing Company, Reading, MA*, 1(98), 9.
- Halligan, S., Altman, D. G., & Mallett, S. (2015). Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: a discussion and proposal for an alternative approach. *European radiology*, 25, 932–939.
- Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45, 171–186.
- Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1), 29–36.
- Hannan, E. J., & Quinn, B. G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2), 190–195.
- Holland, J. (1975). Adaptation in natural and artificial systems, univ. of mich. press. *Ann Arbor*, 7, 390–401.

- Hunt, E. B., Marin, J., & Stone, P. J. (1966). Experiments in induction.
- Hussain, E. (2012). The bi-gamma roc curve in a straightforward manner. *Journal of Basic & Applied Sciences*, 8(2), 309–314.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.
- Jokstad, A., Braegger, U., Brunski, J. B., Carr, A. B., Naert, I., & Wennerberg, A. (2003). Quality of dental implants. *International dental journal*, 53(S6P2), 409–443.
- Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4, 87–112.
- Lee, J.-H., Lee, J.-B., Kim, M.-Y., Yoon, J.-H., Choi, S.-H., & Kim, Y.-T. (2016). Mechanical and biological complication rates of the modified lateral-screw-retained implant prosthesis in the posterior region: an alternative to the conventional implant prosthetic system. *The journal of advanced prosthodontics*, 8(2), 150–157.
- Lee, J.-H., Lee, J.-B., Park, J.-I., Choi, S.-H., & Kim, Y.-T. (2015). Mechanical complication rates and optimal horizontal distance of the most distally positioned implant-supported single crowns in the posterior region: A study with a mean follow-up of 3 years. *Journal of Prosthodontics*, 24(7), 517–524.
- Legendre, A. M. (1805). *Mémoire sur les opérations trigonométriques: dont les résultats dépendent de la figure de la terre*. Number 1. F. Didot.
- Liu, Y. (2007). Robust and misspecification resistant model selection in regression models with information complexity and genetic algorithms.
- LTD., B. D. S. (2024). Dental implants. Accessed: 24-Dec-2024.
- Luis-Delgado, J., Martinez-Alajarin, J., & Tomas-Balibrea, L. (2003). Classification of marble surfaces using wavelets. *Electronics Letters*, 39(9), 1.

- Luque, A., Carrasco, A., Martín, A., & de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91, 216–231.
- Mallat, S. (1999). A wavelet tour of signal processing.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.
- Mitchell, W. J. (1996). *City of bits: space, place, and the infobahn*. MIT press.
- Moraschini, V., Poubel, L. d. C., Ferreira, V., & dos Sp Barboza, E. (2015). Evaluation of survival and success rates of dental implants reported in longitudinal studies with a follow-up period of at least 10 years: a systematic review. *International journal of oral and maxillofacial surgery*, 44(3), 377–388.
- Nelder, J. A., & Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3), 370–384.
- Papaspyridakos, P., Bordin, T. B., Kim, Y.-J., El-Rafie, K., Pagni, S. E., Natto, Z. S., Teixeira, E. R., Chochlidakis, K., & Weber, H.-P. (2020). Technical complications and prosthesis survival rates with implant-supported fixed complete dental prostheses: a retrospective study with 1-to 12-year follow-up. *Journal of Prosthodontics*, 29(1), 3–11.
- Provost, F., & Domingos, P. (2003). Tree induction for probability-based ranking. *Machine learning*, 52, 199–215.
- Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
- Rechenberg, I. (1973). Evolutionsstrategie stuttgart. *RICE, JA (1995) Math*.
- Schwarz, G. (1978). Estimating the dimension of a model annals of statistics, 6, 461-464. *MathSciNet zbMATH*.
- Selver, M. A. (2014). Segmentation of abdominal organs from ct using a multi-level, hierarchical neural network strategy. *Computer methods and programs in biomedicine*, 113(3), 830–852.

- Shibata, Y., Yamamoto, T., Takano, S., Katayama, W., Takeda, T., & Matsumura, A. (2009). Direct comparison of thallium-201 and technetium-99m mibi spect of a glioma by receiver operating characteristic analysis. *Journal of clinical neuroscience*, 16(2), 264–269.
- Takeuchi, K. (1976). Distribution of an information statistic and the criterion for the optimal model. *Mathematical Science*, 153, 12–18.
- Tang, L., Du, P., & Wu, C. (2010). Compare diagnostic tests using transformation-invariant smoothed roc curves. *Journal of statistical planning and inference*, 140(11), 3540–3551.
- van Endem, M. H. (1971). An analysis of complexity. *Amsterdam: Mathematisch Centrum*, 91, Amsterdam.
- Vapnik, V., & Lerner, A. Y. (1963). Recognition of patterns with help of generalized portraits. *Avtomat. i Telemekh*, 24(6), 774–780.
- Vardhan, R. V., Pundir, S., & Sameera, G. (2012). Estimation of area under the roc curve using exponential and weibull distributions. *Bonfring International Journal of Data Mining*, 2(2), 52.
- Wei, L., Yang, Y., & Nishikawa, R. M. (2009). Microcalcification classification assisted by content-based image retrieval for breast cancer diagnosis. *Pattern recognition*, 42(6), 1126–1132.
- Whitley, D. S. (1994). By the hunter, for the gatherer: art, social relations and subsistence change in the prehistoric great basin. *World archaeology*, 25(3), 356–373.
- Williams, L., Bozdogan, H., & Aimon-Smith, L. (1995). Inference problems with equivalent models. In *Inference problems with equivalent models* (279–314). Advanced Structural Equation Modeling Techniques.
- Zou, K. H., Hall, W. J., & Shapiro, D. E. (1997). Smooth non-parametric receiver operating characteristic (roc) curves for continuous diagnostic tests. *Statistics in medicine*, 16(19), 2143–2156.